

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES



PROYECTO DE GRADO
NIVEL LICENCIATURA

SISTEMA DE MONITOREO MASIVO DE PULSO CARDIACO
Y SATURACIÓN DE OXÍGENO EN LA SANGRE PARA EL
CENTRO DE SALUD "VILLA ARMONÍA"

POR: ISAAC ALBARO APAZA RAMIREZ

TUTOR: LIC. EDDER TOMAS JURADO MOYA

LA PAZ – BOLIVIA

2022

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

Proyecto de Grado

**SISTEMA DE MONITOREO MASIVO DE PULSO CARDIACO Y
SATURACIÓN DE OXÍGENO EN LA SANGRE PARA EL CENTRO
DE SALUD "VILLA ARMONIA"**

Presentado por: Isaac Albaro Apaza Ramírez

Para optar al grado académico de licenciado en Electrónica y
Telecomunicaciones

Nota numeral:

Nota literal:

Ha sido

Lic. Julia Torrez Soria

Directora de la carrera de Electrónica y Telecomunicaciones

Tutor: Lic. Edder Tomas Jurado Moya

Tribunal: Ing. José Arturo Marín Thames

Tribunal: Lic. Juan Carlos Gutiérrez Yujra

Tribunal: Ing. Edwin Ibarra García

DEDICATORIA

A Dios por guiarme siempre por un buen camino y seguir adelante.

A mis padres, Juan Apaza Laura y Cristina Ramirez Salas, por brindarme su apoyo incondicional en todos estos años dándome la oportunidad de estudiar una carrera, viviendo juntos alegrías y tristezas que nos da la vida y sobre todo por brindándome su confianza.

A mi gran amigo Jaime Cortez, mi hermano que siempre confió en mí con sus grandes consejos y enseñanzas.

AGRADECIMIENTOS

A mi casa de estudios mayor la UMSA

UNIVERSIDAD MAYOR DE SAN
ANDRES

Quiero agradecer a Dios por el regalo de la vida y con ello la oportunidad de alcanzar metas y objetivos trazados.

A mis padres, hermanos, sobrinos quienes me brindan su cariño y compañía y la fortaleza de seguir hacia adelante.

Al Licenciado Edder Tomas Jurado Moya docente de la carrera Electrónica y Telecomunicaciones por la asesoría, la ayuda y los consejos durante el desarrollo del proyecto.

A todos los docentes de mi carrera en especial a mis tribunales:

Ing. Jose Arturo Marin Thames, Lic. Edwin Ibarra García y a quien en vida fue Lic. Juan Carlos Gutiérrez viva por siempre en nuestros corazones, gracias por su conocimiento, orientación, paciencia y sus correcciones realizadas que fueron fundamentales para la culminación de este proyecto.

ÍNDICE

RESUMEN.....	v
CAPITULO 1.....	1
MARCO REFERENCIAL	1
1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 OBJETIVOS	3
1.3.1 OBJETIVO GENERAL.....	3
1.3.2 OBJETIVOS ESPECÍFICOS.....	3
1.4 JUSTIFICACIÓN	3
1.4.1 JUSTIFICACIÓN ACADÉMICA	3
1.4.2 JUSTIFICACIÓN SOCIAL.....	3
1.4.3 JUSTIFICACIÓN TECNOLÓGICA.....	3
1.5 DELIMITACIONES	4
1.5.1 DELIMITACIÓN ESPACIAL	4
1.5.2 DELIMITACIÓN TEMPORAL.....	4
1.5.2 DELIMITACIÓN TEMATICA.....	4
CAPITULO II	5
MARCO TEORICO	5
2.1 CONCEPTOS BIOLÓGICOS ELEMENTALES	5
2.1.1 EL OXÍGENO.....	5
2.1.2 SUMINISTRO DE OXÍGENO EN EL CUERPO HUMANO.....	5
2.1.3 CANTIDAD DE OXÍGENO QUE CAPTA EL SER HUMANO.....	7
2.1.4 LA SANGRE	7
2.1.5 COMPOSICIÓN DE LA SANGRE.....	8
2.1.6 FUNCIONES PRINCIPALES DE LA SANGRE	9
2.1.7 GLÓBULOS ROJOS	9
2.2 OXÍMETRO.....	10
2.3 OXÍMETRÍA COMO TÉCNICA DE DIAGNÓSTICO BIOMÉDICO.....	11

2.4 OXIMETRÍA DE PULSO	13
2.5 METODOLOGIA DE OXIMETRÍA DE PULSO	14
2.6 SENSOR DE ONDA DEL OXIMETRO.....	16
2.7 AMPLIFICACIÓN Y CONVERSIÓN DE CORRIENTE A TENSIÓN DE LA SEÑAL DE FOTODETECTOR	18
2.8 SEPARACIÓN DE LA COMPONENTE DE LA SEÑAL ARTERIAL	19
2.9 DETERMINACIÓN DE LA SEÑAL DE OXÍGENO (SPO2)	24
2.10 LENGUAJE DE PROGRAMACION DE ALTO NIVEL	25
2.11 VISUAL BASIC.NET.....	25
2.11.1 DIFERENCIAS DE VISUAL BASIC	25
2.11.2 TIPOS DE DATOS	26
2.11.3 DATAGRIDVIEW CON VISUAL BASIC .NET.....	27
2.11.4 CONDICIONALES Y BUCLES	28
2.11.5 PROCEDIMIENTOS Y FUNCIONES	29
2.11.6 FORMULARIO Y CUADRO DE HERRAMIENTAS.....	29
2.11.7 CONTROLES DE USUARIO	29
2.12 BUNIFU FRAMEWORK.....	29
2.13 SOCKETS EN VB.NET	30
2.14 ARDUINO	32
2.14.1 CARACTERISTICAS DEL IDE DE ARDUINO	32
2.15 PROTOCOLO DE INTERNET VERSIÓN 4 IPV4	34
2.16 PROTOCOLO TCP	35
2.17 WIFI	35
2.18 SERVIDOR.....	35
2.19 RS-232	36
2.20 MICROPROCESADOR Y MICROCONTROLADOR.....	37
2.20.1 MICROPROCESADOR	37
2.20.2 MICROCONTROLADOR	38
2.21 ESP-32	38
2.21.1 CARACTERISTICAS DEL ESP32.....	39
2.22 SENSOR MAX30102	41

2.22.1 CARACTERISTICAS DEL SENSOR MAX30102.....	41
CAPITULO III.....	43
MARCO PRÁCTICO.....	43
3.1 DESCRIPCIÓN GENERAL DEL SISTEMA	43
3.2 DESCRIPCIÓN DEL CIRCUITO DEL SISTEMA	44
3.3 PASOS A SEGUIR PARA TRABAJAR CON LA PLACA DE DESARROLLO ESP32.....	46
3.4 INSTALACIÓN DE LIBRERÍAS NECESARIAS EN ARDUINO IDE.....	51
3.5 DISEÑO DEL CIRCUITO ESQUEMÁTICO E IMPRESO EN PROTEUS.....	52
3.6 INTERFAZ DE PC PARA EL PROGRAMA DE MONITOREO.....	53
3.7 DIAGRAMA DE FLUJO	56
3.7.1 DIAGRAMA DEL PROGRAMA ARDUINO.....	56
3.7.2 DIAGRAMA DEL PROGRAMA DE LA PC.....	59
CAPITULO IV	61
ANÁLISIS DE COSTOS	61
4.1 COSTO DE MATERIALES	61
4.2 COSTOS DE OXÍMETROS COMERCIALES USADOS EN CENTROS DE SALUD	61
CAPITULO V.....	63
CONCLUSIONES Y RECOMENDACIONES.....	63
5.1 CONCLUSIONES	63
5.2 RECOMENDACIONES.....	63
BIBLIOGRAFIA.....	65
ADJUNTOS	68
CÓDIGO FUENTE DE LA APLICACIÓN DE CONFIGURACIÓN EN VB.NET..	68
ELABORACIÓN DEL PCB CORRESPONDIENTE.....	73

ÍNDICE DE FIGURAS

Figura 1: Oxímetro.	13
Figura 2: Esquema de medición de oximetría.	16
Figura 3: Técnicas de oximetría por pulso.	17
Figura 4: Absorción de luz.	18
Figura 5: Elementos funcionales básicos de un receptor.	18
Figura 6: Diagrama de bloques sistema de comprensión.	19
Figura 7: Modelo de Parámetros de Pulso Arterial.	20
Figura 8: Segmentación de la Señal a partir de sus mismos extremos.	22
Figura 9: Espectro de frecuencias de un segmento de pulso arterial.	23
Figura 10: Tabla informativa aplicando datagridview.	27
Figura 11: IDE de Arduino.	33
Figura 12: Detalle de Una Dirección IPV4.	34
Figura 13: Conector RS-232 (DE-9 Macho).	36
Figura 14: ESP32.	39
Figura 15: Sensor MAX30102.	42
Figura 16: Diagrama básico del sistema de monitoreo.	43
Figura 17: Diagrama general del dispositivo.	44
Figura 18: Diagrama de bloques del circuito principal.	45
Figura 19: ESP Wroom 32 Dev. Kit V1.	45
Figura 20: ESP32 diseñado en Proteus.	46
Figura 21: IDE de Arduino en escritorio.	46
Figura 22: Frameworks de desarrollo oficial para el ESP32.	47
Figura 23: Gestor de URLs Adicionales de Tarjetas del IDE de Arduino.	48
Figura 24: Gestor de tarjeta del IDE de Arduino.	48
Figura 25: Tarjetas para el módulo ESP32.	49
Figura 26: Selección del módulo ESP32.	49
Figura 27: Compilación y grabado.	50
Figura 28: Verificado y subido al ESP32 exitosamente.	50
Figura 29: Incluir bibliotecas al IDE de Arduino	51
Figura 30: Gestor de librerías del IDE de Arduino.	51
Figura 31: Simulación del proyecto.	52
Figura 32: Esquema del proyecto.	53
Figura 33: PC programa.	54
Figura 34: PC programa en modo oscuro.	54
Figura 35: Recepción de datos en el PC programa modo oscuro.	55
Figura 36: Recepción de datos en el PC, programa modo claro.	55
Figura 37: Diagrama de flujo del programa de Arduino.	58
Figura 38: Diagrama de flujo del programa correspondiente a la PC.	60

RESUMEN

El presente proyecto titulado “Sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre para el centro de salud Villa Armonía”, nace de una necesidad en el centro de salud de Villa Armonía el cual no cuenta con un equipo necesario para poder realizar mediciones de oxígeno en la sangre esto a raíz de una pandemia que se presentó a nivel mundial poniendo en notoriedad la deficiencia en la atención a pacientes que presentaban síntomas de COVID-19.

En base a investigaciones y antecedentes de proyectos relacionados con mediciones de saturación de oxígeno en la sangre se tomaron en cuenta varios datos, que nos permitieron poder desarrollar el diseño del dispositivo tanto como el sistema de monitoreo masivo.

Es por ello que se ha trabajado para desarrollar tanto la parte software como el diseño hardware del dispositivo médico encargado de calcular y estudiar las incógnitas citadas en el capítulo 1 marco referencial, tomando en cuenta teorías que se presenta el documento en el capítulo 2 marco teórico. Teniendo como resultado el producto de un dispositivo medico al cual se lo nombra como “SUMAÑA-SUNQU” los cálculos, diseños y desarrollo se detallan en el capítulo 3 marco practico.

Con el diseño de una sencilla interfaz para el monitoreo, teniendo la capacidad de registrar mediciones a distintos pacientes o aun grupo masivo que atienda el centro de salud de Villa Armonía, con todos los registros realizados. Consiguiendo así una monitorización entre el médico y el paciente completa, evitando provocar riesgos de salud innecesarios.

En la parte final del presente proyecto, se expone la conclusión del trabajo logrando cumplir con las soluciones a los problemas planteados, con el dispositivo también se propone mejoras en el área de salud, permitiendo contar con un equipo desarrollado de comunicaciones inalámbricas para satisfacer al usuario, incluyendo la opción de monitorización.

CAPITULO 1

MARCO REFERENCIAL

1.1 ANTECEDENTES

En la búsqueda de referencias de proyectos o equipos implementados de pulsometría con un sistema de monitoreo en centros de salud se han podido evidenciar investigaciones, proyectos y equipos con algunas similitudes los cuales servirán como antecedentes y referencia.

La técnica nace a través del trabajo de Matthes que desarrolló el primer dispositivo en 1935, siendo un dispositivo de longitud de onda que permitía medir la saturación de oxígeno con filtros rojos y verdes. Siendo este el primer dispositivo para la medición de la saturación de oxígeno.

Se ensambló el primer oxímetro de medición en 1964 por Shaw, usando ocho longitudes de onda de luz, comercializado por Hewlett Packard, su uso fue limitado a funciones pulmonares y a laboratorios de sueño, esto por su particularidad en su costo y tamaño.

La pulsioximetria fue desarrollada en su plenitud en el año 1974, por Takuo Aoyagi y Michio Kishi, bioingenieros de Nihon.

En la actualidad en Bolivia la pulsioximetria es llevada a cabo en pacientes con afecciones pulmonares y cardiacas con la intervención del plantel médico que recurre al oxímetro como herramienta el cual ayuda a visualizar el estado en que se encuentra el pulso cardiaco del paciente a través de un indicador, el médico de turno o la enfermera toma nota en un determinado tiempo (Rondas o Visitas Médicas) verificando la situación cardiaca y su nivel de oxígeno en la sangre.

No se tiene constancia aun, que en algún establecimiento médico de Bolivia cuente con un sistema de monitoreo que sea capaz de monitorear el pulso cardiaco y medir los niveles de oxígeno en la sangre, sin la necesidad que el paciente sea controlado personalmente por un médico especialista.

1.2 PLANTEAMIENTO DEL PROBLEMA

En la actualidad la sociedad en general se ha visto envuelta en un problema de salud inminente, el cual ha sido provocado por el virus COVID-19, el cual es una infección de tipo respiratoria que causa diferentes problemas en los pacientes que lo padecen.

Se ha observado que uno de los síntomas frecuentes es el problema de la respiración, implicando directamente en la saturación de oxígeno en la sangre.

También se ha observado que monitorear individualmente la saturación de oxígeno en la sangre, así como el pulso cardiaco de manera individual dificulta la aceleración de trabajo del personal médico. Por otra parte, también se observa que el monitoreo masivo en pacientes aún no ha sido implementado en ningún centro de salud local, es más se observa que muchos centros de salud de primer nivel no cuentan con oxímetros ni otros insumos necesarios para brindar un mejor servicio.

Por otra parte, se observa que el monitoreo de pulso cardiaco y saturación de oxígeno en la sangre puede servir como diagnóstico para varias enfermedades como ser obesidad y problemas cardiovasculares.

También se debe hacer notar que para el monitoreo masivo se debe tener en cuenta que el cableado debe ser el mínimo ya que el exceso de cables es perjudicial tanto para pacientes como personal médico y de servicio.

Las situaciones descritas anteriormente plantean un problema de investigación el cual debe ser resuelto, para lo cual se deberá tener en cuenta los medios correspondientes para que dicho problema sea resuelto y de una solución a la sociedad que lo requiere. Para lo cual se plantea las siguientes preguntas:

¿Cómo diseñar un sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre usando módulos de comunicación inalámbricos como medio de transmisión?

¿Cuáles serán las características técnicas y económicas del sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre usando módulos inalámbricos como medio de transmisión?

Preguntas que serán respondidas en el desarrollo del proyecto, y dando la solución a los problemas planteados.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre para el centro de salud "Villa Armonía".

1.3.2 OBJETIVOS ESPECÍFICOS

- Determinar las características técnicas del sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre.
- Determinar el tipo y las características del oxímetro en el proyecto.
- Elaborar el firmware del microcontrolador.
- Diseñar el software para ajustes y configuración de parámetros.
- Implementar el prototipo final con la corrección de errores.

1.4 JUSTIFICACIÓN

1.4.1 JUSTIFICACIÓN ACADÉMICA

El presente proyecto aplica los conocimientos adquiridos durante toda la formación universitaria, dando una solución a un determinado problema que implica tanto conocimientos teóricos como prácticos.

1.4.2 JUSTIFICACIÓN SOCIAL

El proyecto se genera a partir de una necesidad de la sociedad, dando una propuesta que será de gran ayuda a la sociedad y también tendrá un costo reducido.

1.4.3 JUSTIFICACIÓN TECNOLÓGICA

En el transcurso de los años la tecnología de los microprocesadores y microcontroladores ha ido evolucionando teniendo estas muchas más funciones y adheridos que le dan un gran

poderío a la hora de dar soluciones a problemas de la sociedad. Es por este motivo que el uso de un microcontrolador que implementa funciones de conexión a red inalámbrica hace posible una solución tecnológica acorde al tiempo actual, facilitando la conexión masiva de varios equipos a un servidor.

1.5 DELIMITACIONES

1.5.1 DELIMITACIÓN ESPACIAL

El sistema de monitoreo presente proyecto se ve limitado a la implementación del prototipo en centros de salud de segundo nivel y que cuenten con disponibilidad de conexión máxima del enrutador WiFi o una conexión directa de internet, por lo que su alcance es condicionado.

1.5.2 DELIMITACIÓN TEMPORAL

Por el tiempo de diseño, el proyecto se presentará como prototipo, realizando pruebas en diferentes pacientes, y comparándolas con un instrumento patrón.

1.5.2 DELIMITACIÓN TEMATICA

Para el monitoreo, uso del oxímetro en unidades médicas se debe emplear en el universo de pacientes con patologías como ser, personas con asma, pacientes con cáncer de pulmón, también personas con enfermedad pulmonar obstructiva crónica (EPOC), y el COVID-19 que con este último se llega tener mayor complicación en los niveles de oxígeno son algunos de los problemas de salud que pueden hacer que los niveles de oxígeno bajen.

CAPITULO II

MARCO TEORICO

2.1 CONCEPTOS BIOLOGICOS ELEMENTALES

2.1.1 EL OXIGENO

El oxígeno es un gas que su cuerpo necesita para funcionar bien. Sus células necesitan oxígeno para producir energía. Sus pulmones absorben el oxígeno del aire que respira. Luego ingresa a la sangre desde los pulmones y viaja a sus órganos y tejidos del cuerpo.

Ciertas afecciones médicas pueden reducir sus niveles de oxígeno en la sangre. Un bajo nivel de oxígeno en la sangre puede hacer que se sienta sin aliento, cansado o confundido. También puede dañar su cuerpo. La terapia con oxígeno puede ayudarle a obtener más oxígeno. Existen diferentes tipos de dispositivos que pueden proporcionarle oxígeno, algunos usan tanques de oxígeno líquido o gaseoso. Otros usan un concentrador de oxígeno, el que extrae oxígeno del aire, se lo puede obtener a través de un tubo nasal (cánula), una máscara o una cámara de oxígeno.

2.1.2 SUMINISTRO DE OXIGENO EN EL CUERPO HUMANO

La cardiopatía cianótica es un trastorno congénito que afecta al flujo de sangre que va hacia y desde los pulmones. Como resultado, la sangre que se bombea a través del cuerpo contiene una cantidad insuficiente de oxígeno. La característica más obvia de la enfermedad cardíaca cianótica es un matiz azulado en los labios y la piel, una clara señal de que no hay suficiente oxígeno en el torrente sanguíneo. Muchos casos de cardiopatía cianótica son leves y pueden tratarse con medicamentos y cirugía. Los problemas más graves pueden requerir un trasplante de corazón para evitar complicaciones mortales.¹

¹ MEDLINE PLUS (2021). *Terapia con Oxígeno*. Recuperado de <https://medlineplus.gov/spanish/oxygentherapy.html>

Hay muchos tipos de enfermedad cardíaca cianótica, clasificados por el tamaño y la localización de defectos específicos. La forma más común de la enfermedad se conoce como tetralogía de Fallot, una condición compuesta de cuatro problemas distintos en el corazón. La tetralogía de Fallot implica una obstrucción de la válvula pulmonar, que normalmente transporta sangre pobre en oxígeno desde el corazón hacia los pulmones, lo que puede ser repuesta. El suministro de oxígeno se reduce aún más debido a que un agujero en el corazón, llamado un defecto del tabique, permite que la sangre se utiliza para volver a entrar directamente en el torrente sanguíneo como se aprecia en la figura 3. La mayoría de otros tipos de cardiopatía cianótica incluyen defectos cardíacos similares que afecten la re-oxigenación de la sangre.

En los casos leves de la enfermedad cardíaca cianótica, sólo una pequeña cantidad de sangre nunca llega a los pulmones. La nariz y los labios de un niño que ha nacido con una cardiopatía cianótica leves pueden ser ligeramente de color azul, y él o ella pueden tener dificultades para inhalar profundamente. Los síntomas tienden a disminuir a medida que el bebé crece, aunque un niño con la enfermedad cardíaca puede no ser capaz de participar en actividades regulares sin ser excesivamente fatigado. La condición a veces conduce a un retraso en el desarrollo mental y físico.

Un médico puede diagnosticar la enfermedad cardíaca cianótica, analizando los resultados de las radiografías de tórax, tomografías computarizadas de tomografía, ecocardiogramas y pruebas de sangre. La mayoría de los pacientes son remitidos a los cirujanos cardíacos que pueden eliminar los bloqueos y reparar los defectos septales. Un cirujano también puede reemplazar las válvulas dañadas o arterias artificiales para asegurar el flujo sanguíneo adecuado. Generalmente, los pacientes deben tomar medicamentos de prescripción y recibir chequeos regulares durante muchos años después de la cirugía para los problemas del corazón seguro no persiste.

Cardiopatía cianótica grave implica un defecto importante en el que muy poco o nada de sangre llega a los pulmones. Un bebé con esta afección puede ser indiferente y tener una respiración muy superficial durante el parto. Se deben tomar medidas de tratamiento de

emergencia para salvar la vida de un bebé en el caso de un defecto importante. El paciente se coloca en un respirador por un tubo listo para la cirugía.²

2.1.3 CANTIDAD DE OXIGENO QUE CAPTA EL SER HUMANO

En 1 litro de aire hay un 21% de oxígeno; es decir 210 ml. Una persona de 70 kg, en reposo, respira unos 6 litros de aire cada minuto. Por lo tanto, cada minuto introduce en su cuerpo $6 \times 210 = 1260$ ml de oxígeno. En el estado de reposo citado, el metabolismo de esa persona gasta $3,5 \text{ ml/kg/min} \times 70 \text{ kg} = 245$ ml de oxígeno cada minuto. Por lo tanto, esos 6 litros de aire que espira durante el minuto todavía contienen, una vez usados, $1260 - 245 = 1015$ ml de oxígeno. Si calculamos la concentración de oxígeno del aire espirado, vemos que el resultado es de aproximadamente un 17% en lugar del 21% que tenía antes de ser respirado.

2.1.4 LA SANGRE

La sangre es tejido vivo formado por líquidos y sólidos. La parte líquida, llamada plasma, contiene agua, sales y proteínas. Más de la mitad del cuerpo es plasma. La parte sólida de la sangre contiene glóbulos rojos, glóbulos blancos y plaquetas.

Los glóbulos rojos suministran oxígeno desde los pulmones a los tejidos y órganos. Los glóbulos blancos combaten las infecciones y son parte del sistema inmunitario del cuerpo. Las plaquetas ayudan a la coagulación de la sangre cuando sufre un corte o una herida. La médula ósea, el material esponjoso dentro de los huesos, produce nuevas células sanguíneas. Las células de la sangre constantemente mueren y su cuerpo produce nuevas. Los glóbulos rojos viven unos 120 días y las plaquetas viven cerca de seis. Algunos glóbulos blancos de la sangre viven menos de un día, pero otros viven mucho más tiempo.

Hay cuatro grupos de sangre: A, B, AB y O. Asimismo, la sangre es Rh positivo o Rh negativo. Así, si su tipo de sangre es A, es A positivo o A negativo. Su tipo de sangre es importante si necesita una transfusión de sangre. Y su factor de sangre puede ser

² MEDLINE PLUS, (2021). <https://medlineplus.gov/spanish/oxygentherapy.html>

importante si usted queda embarazada, ya que la incompatibilidad entre su tipo de sangre y el de su bebé puede crear problemas.

Los análisis de sangre como las pruebas de conteo sanguíneo ayudan a los médicos a analizar ciertas enfermedades y afecciones. También ayudan a comprobar la función de los órganos y muestran qué tan bien están funcionando los tratamientos. Algunos problemas de la sangre pueden incluir problemas de coagulación, coágulos sanguíneos y desórdenes plaquetarios. Si pierde mucha sangre, usted puede necesitar una transfusión.³

2.1.5 COMPOSICIÓN DE LA SANGRE

Como todo tejido, la sangre se compone de células y componentes extracelulares (su matriz extracelular). Estas dos fracciones tisulares vienen representadas por:

- Los **elementos formes** —también llamados elementos figurados—: son elementos semisólidos (es decir, mitad líquidos y mitad sólidos) y particulados (corpúsculos) representados por células y componentes derivados de células.
- El **plasma sanguíneo**: un fluido traslúcido y amarillento que representa la matriz extracelular líquida en la que están suspendidos los elementos formes. Este representa un medio isotónico para las células sanguíneas, las cuales sobreviven en un medio que esté al 0,9 % de concentración, como la solución salina, para proporcionar un ejemplo.

Los elementos formes constituyen alrededor del 45 % de la sangre. Tal magnitud porcentual se conoce con el nombre de hematocrito (fracción "celular"), adscribible casi en totalidad a la masa eritrocitaria. El otro 55 % está representado por el plasma sanguíneo (fracción acelular).

Los elementos formes de la sangre son variados en tamaño, estructura y función, y se agrupan en:

³ MEDLINE PLUS (2021).
<https://medlineplus.gov/spanish/blood.html>

- Las **células sanguíneas**, que son los *glóbulos blancos* o *leucocitos*, células que "están de paso" por la sangre para cumplir su función en otros tejidos;
- Los **derivados celulares**, que no son células estrictamente sino fragmentos celulares, están representados por los *eritrocitos* y las *plaquetas*; son los únicos componentes sanguíneos que cumplen sus funciones estrictamente dentro del espacio vascular.⁴

2.1.6 FUNCIONES PRINCIPALES DE LA SANGRE

Como todos los tejidos del organismo la sangre *cumple múltiples funciones* necesarias para la vida como la defensa ante infecciones, los intercambios gaseosos y distribución de nutrientes. Para cumplir con todas estas funciones cuenta con diferentes tipos de células suspendidas en el plasma. Todas las células que componen la sangre se fabrican en la médula ósea. Ésta se encuentra en el tejido esponjoso de los huesos planos (cráneo, vértebras, esternón, crestas ilíacas) y en los canales medulares de los huesos largos (fémur, húmero).

La sangre es un tejido renovable del cuerpo humano, esto quiere decir que la médula ósea se encuentra fabricando, durante toda la vida, células sanguíneas ya que éstas tienen un tiempo limitado de vida. Esta "fábrica", ante determinadas situaciones de salud, puede aumentar su producción en función de las necesidades. Por ejemplo, ante una hemorragia aumenta hasta siete veces la producción de glóbulos rojos y ante una infección aumenta la producción de glóbulos blancos.⁵

2.1.7 GLÓBULOS ROJOS

Los glóbulos rojos son un componente importante de la sangre, su función es transportar oxígeno a los tejidos corporales e intercambiarlo por dióxido de carbono, el cual es transportado y eliminado por los pulmones. Los glóbulos rojos se forman en la médula ósea roja de los huesos, las células madre de la médula ósea roja, llamadas hemocitoblastos, dan lugar a todos los elementos formados en la sangre. Si un

⁴ Wikipedia, la enciclopedia libre, (2021). *La Sangre* Recuperado de https://es.wikipedia.org/wiki/Sangre#Composici%C3%B3n_de_la_sangre

⁵ Ministerio de Salud (2021) *Composición de la Sangre* Recuperado de http://www.msal.gov.ar/disahe/index.php?option=com_content&id=315&Itemid=39

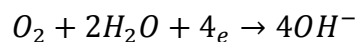
hemocitoblasto se convierte en una célula llamada proeritroblasto, esta se convertirá luego en un nuevo glóbulo rojo, la formación de un glóbulo rojo a partir de un hemocitoblasto toma alrededor de dos días. ¡El cuerpo produce unos dos millones de glóbulos rojos cada segundo! La sangre está compuesta tanto por elementos celulares como líquidos. Si una muestra de sangre se pone a girar en una centrífuga, los elementos formados y la matriz variable de la sangre se pueden separar entre sí, la sangre contiene un 45% de glóbulos rojos, menos de 1% de glóbulos blancos y plaquetas y un 55% de plasma.

2.2 OXIMETRO

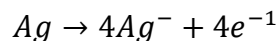
Un **Oxímetro** es un aparato que se emplea para determinar la concentración de oxígeno en una disolución acuosa.

En los Oxímetros más comunes el sensor del oxígeno consta de un ánodo o electrodo de referencia (normalmente de plata) y un cátodo, o electrodo de trabajo (de oro o, menos habitualmente, de platino) situados en el interior de un cilindro vacío que contiene una disolución electrolítica de cloruro de potasio, KCl, y está cerrado por una membrana de teflón permeable al oxígeno. Este electrodo de trabajo se denomina electrodo de Clark porque fue diseñado por el bioquímico norteamericano Leland C. Clark Jr. (1918–2005).

Al sumergir el sensor dentro de una muestra de agua, el oxígeno que contiene disuelto atraviesa la membrana; en el cátodo se produce una reducción de las moléculas de oxígeno, O_2 , a aniones hidroxilo, OH^- :



En el ánodo se produce la semirreacción de oxidación que proporciona los cuatro electrones:



Esta reacción electroquímica produce una corriente eléctrica entre el ánodo y el cátodo cuya intensidad depende de la llegada de oxígeno al cátodo, que a la vez es función del gradiente de concentración de oxígeno disuelto entre las dos caras de la membrana, que a la vez depende únicamente de la concentración de la cara externa (puesto que el cátodo consume el oxígeno que le llega). Por eso la intensidad es directamente

proporcional a la concentración de oxígeno del agua problema. Esta medida se transforma automáticamente en concentración de oxígeno en porcentaje respecto a la saturación o en otros tipos de unidades.

La determinación de la concentración de oxígeno en el agua tiene interés en estudios oceanográficos, en estudios que precisan determinar tasas de fotosíntesis o metabolismo.⁶

2.3 OXIMETRÍA COMO TÉCNICA DE DIAGNÓSTICO BIOMÉDICO

La relativa baja atenuación de luz en tejido biológico en el intervalo espectral entre las longitudes de onda de 650 nm a 1050 nm, hace que esta ventana espectral sea de gran interés para monitoreo en pacientes. La gran ventaja de las técnicas ópticas es que son no invasivas, no ionizantes y no destructivas, no producen daño ni modificación a la muestra o sujeto bajo prueba, esto permite un monitoreo continuo.

Posiblemente, su principal desventaja para efectos de diagnóstico sea que un sujeto, en estado vivo, introduce en las mediciones muchas condiciones adversas (ruido aleatorio), en comparación con las mediciones estables y estáticas hechas in-vitro. Como ejemplo mencionamos la aparición de tipos de ruidos variados presentes en las mediciones debido al funcionamiento del cuerpo humano, órganos vivos, que producen estos efectos temporales incontrolables, irregulares, y muchos impredecibles.

Incluyen espasmos musculares involuntarios y contracciones inesperadas debido a la salud y estado del paciente. Los datos espectroscópicos obtenidos en un espécimen vivo son interferidos por una cantidad mucho más apreciable que aquellos tomados en materiales de composición comparable, como fantasmas (phantoms), o tejido in vitro. Esto es tal vez una de las razones principales por las que las técnicas ópticas se han ido desarrollando e introduciendo lentamente en los protocolos de diagnóstico en pacientes, a pesar de sus grandes ventajas encontradas en los últimos treinta años.

⁶ Wikipedia, la enciclopedia libre, (2020).
<https://es.wikipedia.org/wiki/Ox%C3%ADmetro>

La oximetría es una técnica óptica que permite calcular la cantidad de oxígeno presente en sangre para diagnóstico médico, por ejemplo, en pacientes que sufren de enfermedades respiratorias o dificultades pulmonares crónicas, asma y sus efectos, enfermedades cardiovasculares, problemas de suministro de oxigenación.

El oxímetro se ha incorporado como un dispositivo estándar de monitoreo en cirugía, cuidados intensivos, terapia intermedia, para detectar niveles de hipoxia (falta de suministro de oxígeno). Este método de monitoreo nos ofrece un nuevo enfoque para observar el comportamiento hemodinámico, el uso y control de anestesia, monitoreo de hemoglobina, detección de carboxihemoglobina y metahemoglobina, predicción de la reacción a fluidos (medicamentos, sustancias de contraste); todo esto era antes sólo posible a través de monitoreo invasivo. Adicionalmente, vemos que los oxímetros son útiles y adecuados para monitoreo en pacientes inmovilizados, no responsivos, poco controlados y no cooperativos, como lo son los niños. Además, los sistemas ópticos son una alternativa a los sistemas de imagenología comunes como la resonancia magnética o la tomografía por positrones, al manejar radiación no ionizante, por lo tanto, no dañina.

Los métodos ópticos no son una competencia de las exitosas técnicas magnéticas (no compiten en resolución de profundidad), sino que son técnicas complementarias. Desde los años 1990 son varios los grupos de investigación que han estado trabajando en espectroscopía de tejidos y que han desarrollado sistemas ópticos para hacer imagen de los procesos fisiológicos dinámicos humanos. Varios sistemas se han construido para registrar y evaluar mediciones biológicas cuantitativas explotando las características de las mediciones de absorción en zonas cercanas al pico de absorción del agua.

Otro ejemplo como el de la universidad de Standford donde han desarrollado métodos ópticos que registran los cambios en la oxigenación de la sangre al trans iluminar tejido. Reportan cambios de oxigenación en cerebro debido a traumatismos, a una conmoción, al proceso de pensar; estos cambios pueden ser cuantificados y se pueden obtener sus imágenes. El monitoreo de estos cambios es importante en cirugía para prevenir lesiones cerebrales debido a episodios hipóxicos o isquémicos (isquemia es el sufrimiento celular causado por la disminución transitoria o permanente del riego sanguíneo y consecuente disminución del aporte de oxígeno “hipoxia”). Otro ejemplo del uso de la oximetría es el

de la universidad de Pennsylvania donde han obtenido imágenes de zonas del cerebro durante trabajo cerebral en niños y adultos. Su método es asequible y lograron un dispositivo portátil para analizar y monitorear funcionalidad, o bien defectos funcionales, sin que el paciente tenga que estar en reposo, es decir, puede continuar con sus rutinas diarias.⁷

2.4 OXIMETRÍA DE PULSO

La oximetría de pulso es un método no invasivo que permite la estimación de la saturación de oxígeno de la hemoglobina arterial y también vigila la frecuencia cardiaca y la amplitud del pulso. La presión parcial de oxígeno disuelto en la sangre arterial se denomina PaO_2 . El porcentaje de saturación de oxígeno unido a la hemoglobina en la sangre arterial se denomina SaO_2 y cuando se mide por un oxímetro de pulso, este valor se denomina SpO_2 .



Figura 1: Oxímetro.

Fuente: <https://yoemed.com/producto/oximetro-de-pulso-tamano-adulto/>

La oximetría de pulso se ha convertido en un instrumento indispensable en la atención de pacientes. Incorporada en nuestro medio, pero sin un proceso de capacitación paralelo y al parecer existen vacíos en el conocimiento del mecanismo de funcionamiento, sus limitaciones y aplicaciones en pediatría; situaciones que abordamos más adelante.⁸

⁷ Centro De Investigaciones En Óptica, (2020). *Oximetría* Recuperado de <https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/606/1/15270.pdf>

⁸ Educación Médica Continua, (2021). *Oximetría de Pulso* Recuperado de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1024-06752012000200011

2.5 METODOLOGIA DE OXIMETRÍA DE PULSO

Para la determinación de la saturación de hemoglobina arterial con oxígeno (SpO_2), el oxímetro de pulso o pulsioxímetro usa la espectrofotometría basada en que la oxihemoglobina u hemoglobina oxigenada (HbO_2) y la desoxihemoglobina o hemoglobina reducida (Hb) absorben y transmiten determinadas longitudes de onda del espectro luminoso para la luz roja (640-660nm) y la luz infrarroja (910-940nm). La HbO_2 absorbe más la luz infrarroja y permite el paso de la luz roja; por el contrario, la Hb absorbe más la luz roja (R) y permite el paso de la luz infrarroja (IR). El ratio de la absorción de la luz R e IR mide el grado de oxigenación de la hemoglobina.

Los oxímetros de pulso tienen dos sensores o sondas con diodos emisores de luz (DEL), uno para luz IR y otro para la R, además, de un fotodiodo detector. Para medir el oxígeno los DEL y el fotodiodo detector deben ponerse en puntos opuestos dejando en medio el tejido translucido (pulpejo del dedo, pabellón auricular, etc). El mecanismo que permite la lectura de la oxigenación es que en cada pulsación de la sangre arterial se transmiten valores lumínicos, detectando al mismo tiempo la frecuencia cardiaca. Asumiendo que solo la sangre arterial pulsa a esto se denomina componente arterial pulsátil (CA). La cantidad de luz absorbida cambia de acuerdo a la cantidad de sangre en el lecho tisular y la presencia de HbO_2/Hb . Por otro lado, existe un componente estático (CE) que está formado por los tejidos, huesos, piel y la sangre venosa. La siguiente fórmula muestra como del cociente de la luz R e IR se obtiene la SpO_2 :

$$\frac{(CA \text{ Luz } \frac{R}{CE} \text{ Luz R})}{(CA \text{ Luz } \frac{IR}{CE} \text{ Luz})} = SpO_2$$

El resultado de la anterior fórmula es llevado a algoritmos calibrados que están almacenados en el microprocesador del aparato. Los algoritmos derivan de mediciones en voluntarios sanos sometidos a diferentes concentraciones mezcladas y decrecientes de oxígeno que generalmente son únicas para cada fabricante. Los actuales oxímetros además de la SpO_2 y la frecuencia cardiaca, muestran una onda pletismográfica de la absorción de

luz. La SpO₂ mostrada en la pantalla representa la media de la medición de los últimos 3 a 6 segundos, los datos se actualizan cada 0.5 a 1 segundo.

La precisión y exactitud dependen de las diferentes marcas y estudios realizados que van desde más o menos 10% a menos de 2%, pero en sujetos con saturaciones de oxígeno por encima de 70%; por esto se ha visto que el funcionamiento de los oxímetros disminuyen su precisión grandemente cuando las SpO₂ están por debajo de 70% (esto se debe a que se tiene un número limitado de sujetos para calibrar a bajos niveles de saturación), lo cual llevaría a serias dudas de su interpretación en pacientes muy hipoxémicos sobre todo en grandes alturas; como menciona un autor, el oxímetro puede fallar "cuando más se lo necesita". La Food and Drug Administration (FDA) considera que las mediciones de cada nuevo oxímetro y sensor deben ser comparados con muestras medidas en sangre arterial con un co-oxímetro en un rango de 70 a 100%.

Los sitios del cuerpo que generalmente se usan para medir la SpO₂ son los dedos de la mano, dedo gordo del pie y lóbulo de la oreja. En neonatos y lactantes menores se usan las palmas y plantas. Otros lugares menos frecuentes son la lengua, alas de la nariz y las mejillas.

La SatO₂ representa los gramos de hemoglobina (Hb) que son portadores de oxígeno. Por ejemplo, si la SpO₂ es del 85 % y la concentración de (Hb) es de 15 g/dl, entonces 12,75 g/dl de Hb están portando O, lo que no ocurre con el 2,25 g/dl restante. El aporte y la entrega de O a los tejidos dependen de varios factores, no sólo del porcentaje de SatO₂. Entre ellos se encuentra el contenido de oxígeno disuelto en la sangre PaO₂.

Por otro lado, es importante recordar que los oxímetros de pulso funcionan normalmente en pacientes anémicos, que tienen reducida la cantidad de glóbulos rojos. Hay que considerar que, en pacientes muy anémicos, la saturación de oxígeno puede ser normal, pero hay insuficiente hemoglobina para transportar una cantidad adecuada de oxígeno a los tejidos.

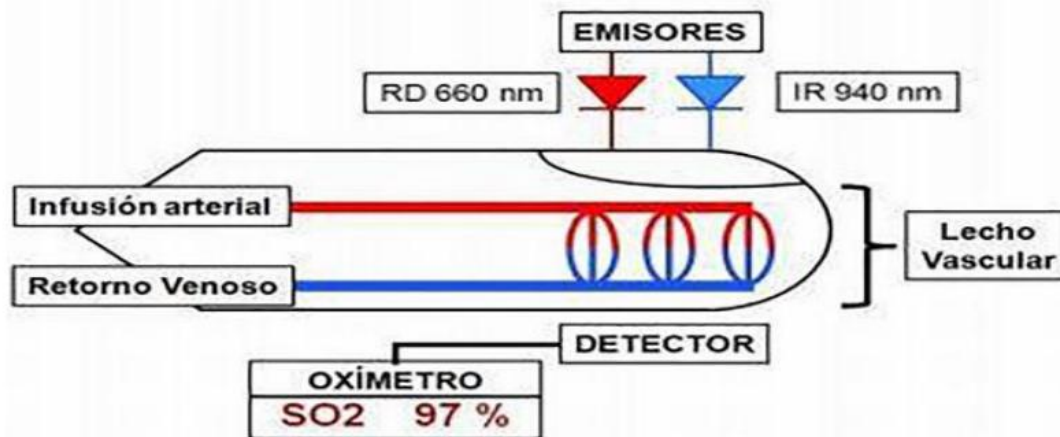


Figura 2: Esquema de medición de oximetría.
Fuente: <http://www.portalbiomedico.com/equipamiento-biomedico/oximetro/oximetria-de-pulsoconceptos.html>

2.6 SENSOR DE ONDA DEL OXIMETRO

El primer concepto que tiene que quedar claro sobre la onda de pulso por pletismografía es que esta no es análoga a la onda de presión sanguínea si no que es una imagen especular de la forma de onda de la intensidad de luz transmitida y medida mediante pletismografía fotoeléctrica. El pletismograma es la representación de la forma de onda de la oxigenación arterial y la detección de la frecuencia del pulso. La frecuencia cardíaca es el número de veces que late el corazón por minuto.

Un corazón normal en reposo late entre 40 y 100 veces por minuto (bpm). A los valores cercanos a los 100 bpm se habla de taquicardia e inferior a los 40 bpm se habla de bradicardia.⁹

El prototipo de pulso es capaz de dar dos tipos diferentes de información:

- La saturación oxígeno medida por oximetría
- La onda pletismografica del volumen sanguíneo

⁹ Somib, (Sociedad Mexicana Biomedica), (2020) Oximetría con pletismógrafo
http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1024-06752012000200011

Recuperado de

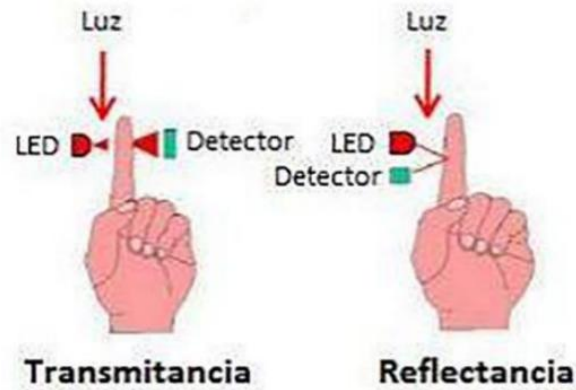


Figura 3: Técnicas de oximetría por pulso.

Fuente: <https://blog.digitexmedical.com/2012/03/30/how-does-a-pulse-oximeter-work/>

El sensor o sonda del oxímetro de pulso consiste precisamente en dos diodos emisores de luz (DEL), uno para la luz roja y otro para la infrarroja, y un fotodiodo detector.

Para mejorar el rendimiento, los DEL y el detector deben colocarse en puntos opuestos de un lugar per fundido que sea translúcido. El fotodiodo mide tres niveles lumínicos diferentes: la luz roja, la luz infrarroja y también la luz ambiente.

Los oxímetros de pulso tienen dos sensores o sondas con diodos emisores de luz (LED), uno para luz Infrarroja y otro para la luz Roja, véase Figura 3, además de un fotodiodo detector. Para medir el oxígeno los diodos leds transmisores y el fotodiodo detector deben ponerse en puntos opuestos dejando en medio el tejido translucido (pulpejo del dedo, pabellón auricular, etc.). El mecanismo que permite la lectura de la oxigenación es que en cada pulsación de la sangre arterial se transmiten valores lumínicos, detectando al mismo tiempo la frecuencia cardiaca. Asumiendo que solo la sangre arterial pulsa a esto se denomina componente arterial pulsátil (CA). La cantidad de luz absorbida cambia de acuerdo con la cantidad de sangre en el lecho tisular y la presencia de HbO₂/Hb. Por otro lado, existe un componente estático (CE) que está formado por los tejidos, huesos, piel y la sangre venosa.¹⁰

¹⁰ S. Cielo (Sociedad Cielo), (2020). *Oxímetro de Pulso* Recuperado de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1024-06752012000200011

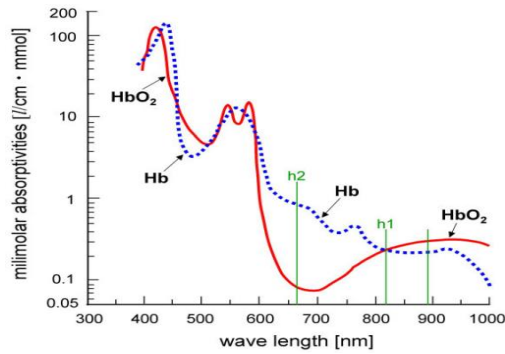


Figura 4: Absorción de luz.

Fuente: https://openi.nlm.nih.gov/detailedresult?img=PMC1851706_1751-0759-1-8-3&req=4

En pocas palabras, un oxímetro tiene unos LED que emiten una luz roja con una longitud de onda de 660 nanómetros y una luz infrarroja con una longitud de onda de 940 nanómetros. En la parte inferior del oxímetro hay un fotodiodo que recibe la luz no absorbida, y así se puede estimar de forma indirecta el nivel de oxígeno en sangre.

2.7 AMPLIFICACIÓN Y CONVERSIÓN DE CORRIENTE A TENSIÓN DE LA SEÑAL DE FOTODETECTOR

El fotodetector es un dispositivo esencial en el bloque receptor de un sistema de comunicaciones ya que recibe una señal óptica y la convierte en señal eléctrica. Una vez que la señal se encuentra en el dominio eléctrico, pasa por etapas posteriores de amplificación y demodulación con las cuales se logra recuperar una versión similar de la información originalmente transmitida.¹¹

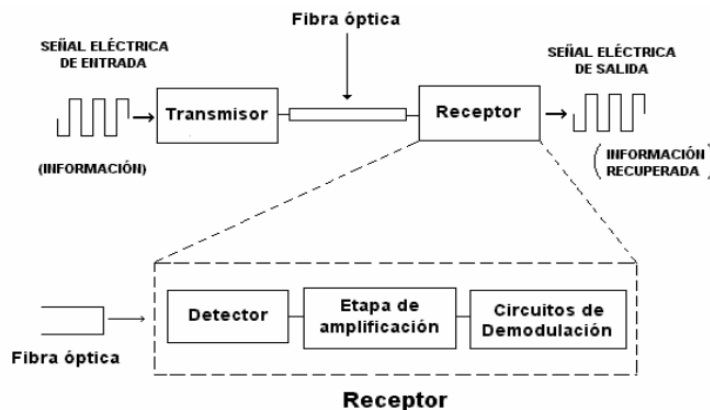


Figura 5: Elementos funcionales básicos de un receptor.

Fuente: https://kipdf.com/universidad-tecnologica-de-la-mixteca_5ac683951723dd1e5009afcf.html

¹¹ Universidad Tecnológica De La Mixteca, (2020). *Medidor de Potencia Óptica* Recuperado de https://kipdf.com/universidad-tecnologica-de-la-mixteca_5ac683951723dd1e5009afcf.html

2.8 SEPARACIÓN DE LA COMPONENTE DE LA SEÑAL ARTERIAL

Los equipos concebidos para la medición de la señal de pulso arterial, diseñados y contruidos en el Centro de Biofísica Médica, (ANGIODIN PD3000) registran solamente períodos cortos de tiempo (aproximadamente 39 segundos). Dentro de las nuevas características que se desean incorporar en los futuros pletismógrafos con procesadores digitales de señales, está la de aprovechar la descripción morfológica de la onda de pulso arterial, por ejemplo, para pronosticar un estado de crisis vasooclusiva en los pacientes sicklémicos y en el seguimiento farmacológico de pacientes con infarto agudo crónico a través de vasodilatadores con acción periférica. Estas aplicaciones requieren del procesamiento, almacenamiento en memoria de largos intervalos de la onda de pulso y transmisión (hacia redes hospitalarias y laboratorios) de altos volúmenes de información (en su generalidad el almacenamiento de más de un estudio).

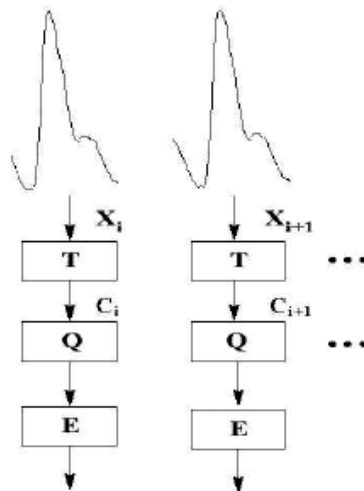


Figura 6: Diagrama de bloques sistema de compresión.

Fuente: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212005000200003

En la Figura 6 se representa de forma esquemática un diagrama en bloques de un sistema de compresión basado en transformadas. La señal de pulso arterial es segmentada en fragmentos de duración N , siendo N la duración del latido más largo. Como no todos los latidos tienen la misma duración, los latidos más cortos se extienden con ceros.

La segmentación se realiza a partir de la detección de los mínimos extremos de cada pulso arterial. Cada fragmento se representa mediante el vector X_i al cual se aplica la transformación ortogonal T . Con ello se consigue una descripción de la señal, vector de coeficientes C_i en un nuevo dominio que resulta más apropiado para su representación de forma eficiente. Posteriormente se procede a la cuantificación Q de los coeficientes transformados C_i .

La transformada ortogonal T logra describir la señal en un nuevo dominio en el que resulta más eficiente la representación de la señal. La cuantificación Q en el dominio transformado introduce una transformación no lineal en la señal y es donde se produce la pérdida de información. En algunos casos, luego de efectuar el proceso de cuantificación, se realiza la codificación por entropía E ; este es un procedimiento reversible que no produce pérdidas y que reduce la redundancia originada por la uniformidad de la función densidad de probabilidad de los coeficientes cuantificados. Las etapas que conforman el algoritmo de compresión serán explicadas a continuación.

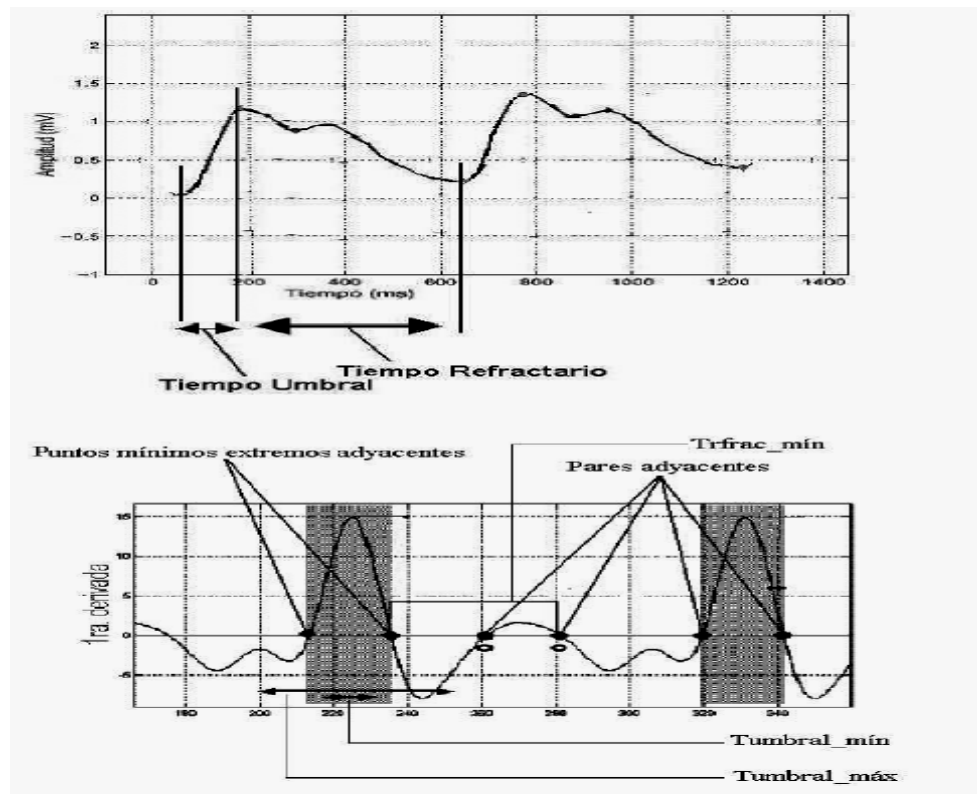


Figura 7: Modelo de Parámetros de Pulso Arterial.

Fuente: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212005000200003

El pulso arterial periférico es fundamentalmente consecuencia de los cambios del volumen de sangre debido a la actividad cardíaca, además de la propia superficie arterial en su propiedad contráctil y relajante. Existe una correspondencia temporal entre las señales de electrocardiografía y pulso arterial periférico. Es de esta relación en la que los parámetros relacionados con la sístole y la diástole en la señal de ECG se reflejan unívocamente en la señal de pulso arterial como parámetros que siempre van acompañados en cada respuesta pulsátil. Los tiempos definidos en la en la Figura 7 se definen como el intervalo en el que se produce la contracción ventricular (tiempo umbral) y el intervalo en el que existe ausencia de actividad muscular cardíaca (tiempo refractario).

Es importante pasar la señal por un filtro pasa banda, en forma tal que se puedan eliminar las componentes indeseadas. Por ejemplo, las variaciones de la línea base y componentes de alta frecuencia, que no tienen origen en el corazón, sino que provienen de la actividad eléctrica de los músculos del tórax (ruido muscular o electromiograma) y el ruido producido por la instrumentación. Para poder realizar una detección automática del pulso lo más óptimo posible, que se puedan descartar los posibles artefactos en la señal durante el proceso de segmentación, se utiliza el modelo de parámetros del pulso arterial. Este modelo hace uso de la primera derivada de la señal para la localización de los mínimos extremos. Se proceden a seleccionar los pares de puntos mínimos extremos adyacentes que cumplen con la condición del tiempo umbral predefinido.

Posteriormente se realiza una comparación entre pares de mínimos adyacentes, verificando la condición del tiempo refractario predefinido. Con la comparación entre todos los pares de muestras finaliza el proceso, quedando almacenados los índices que representan los mínimos extremos que se corresponden con la señal original (punto de inicio y final de cada segmento Figura 8).

Con esta información se conforma una matriz de dos dimensiones que almacena por columnas los segmentos de la señal original. Una vez que se hayan almacenado todos los valores pertenecientes a un segmento, éste se rellena con cero hasta alcanzar la longitud del segmento de mayor duración N .

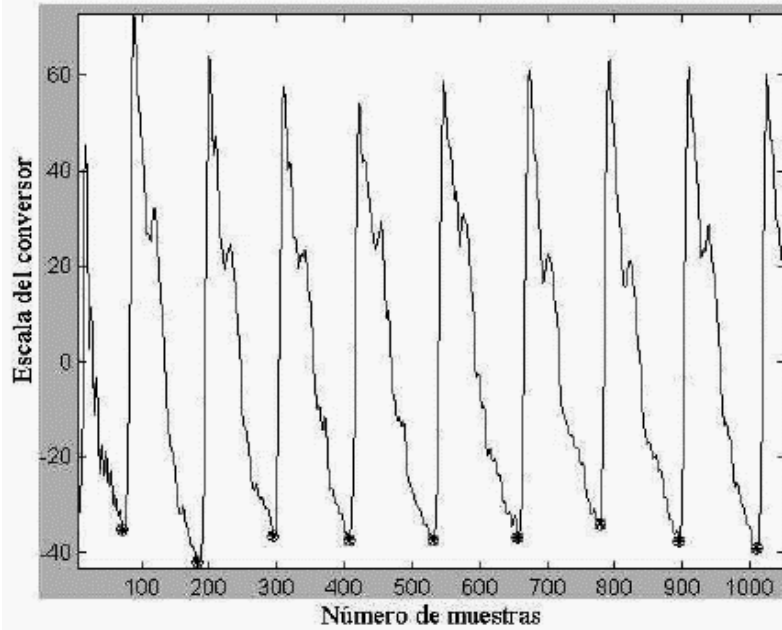


Figura 8: Segmentación de la Señal a partir de sus mismos extremos.
Fuente: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212005000200003

Para cada segmento de la matriz se efectúa la transformada. Para este proceso se utiliza la transformada discreta de Fourier (DFT, Discrete Fourier Transform), que utiliza funciones senos y cosenos para representar la señal y la transformada discreta del Coseno (DCT, Discrete Cosine Transform), que utiliza solamente funciones cosenos. La selección de ambas transformadas está fundamentada en los algoritmos eficientes de cálculo basados en la transformada rápida de Fourier (FFT, Fast Fourier Transform), que se proporcionan comercialmente y que posibilitan disminuir los costos computacionales en el proceso de compresión. Antes de efectuar el proceso de transformación en la DCT, la señal en el dominio del tiempo se hace simétrica con respecto a ella 0, lo que significa que una señal compuesta de 0 a M muestras realiza un duplicado o espejo de la misma desde 1 hasta M.

Realizado el proceso de transformación por cada segmento del pulso, se procede a la estimación de los coeficientes de mayor energía. Como se observa en la Figura 9, en el espectro de frecuencias de las transformadas DFT y DCT de un segmento de la señal de pulso arterial, las componentes frecuenciales de más bajo orden poseen mayor energía, por lo que si se logra agrupar un número p de funciones base asociado a estas componentes frecuenciales es posible reconstruir la señal con el mínimo error posible. Como resultante

final el número de funciones base asociado a cada segmento es variable, lo que trae una ligera mejora cuando se realicen los cálculos finales del índice de razón de compresión.

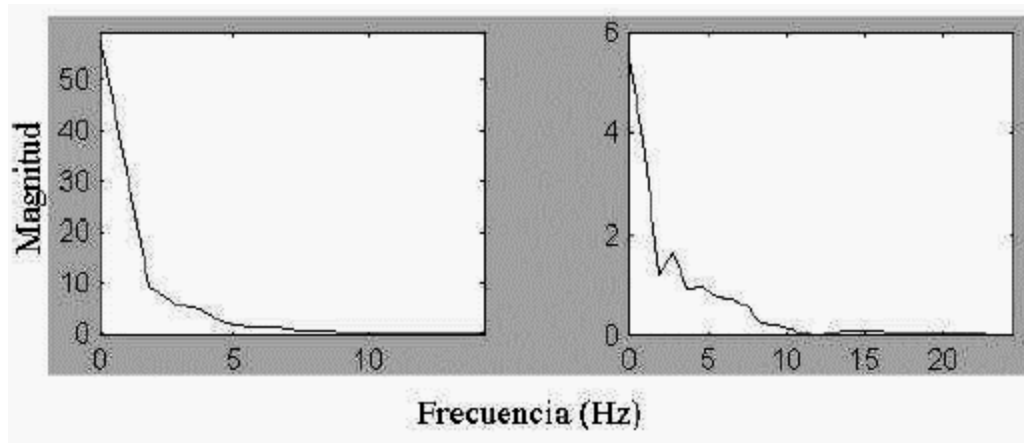


Figura 9: Espectro de frecuencias de un segmento de pulso arterial.

Fuente: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212005000200003

La parte real de los coeficientes transformados, pertenecientes al segmento, se eleva al cuadrado para la cuantificación del contenido energético de cada coeficiente, de manera que se pueda determinar entre ellos el de mayor valor. Se selecciona un valor umbral medio, tomando como referencia el coeficiente de mayor energía, que permita seleccionar un número de coeficientes que cumplan la condición de que su energía sea mayor o igual a este valor umbral. Estos coeficientes seleccionados representan un número p de funciones base. Un reducido número de estas funciones permite elevadas relaciones de compresión. Sin embargo, esto puede provocar un efecto de distorsión más notable sobre la señal reconstruida. Los coeficientes seleccionados son almacenados en un vector común a todos los segmentos para lo cual se almacena el número de coeficientes por segmento y su duración. Estos parámetros resultan imprescindibles para la reconstrucción de la señal. El proceso concluye con la estimación de coeficientes de todos los segmentos transformados de la señal, quedando en un vector la información de esta primera fase de compresión para su posterior cuantificación.¹²

¹² Universidad, Ciencia Y Tecnología, (2020). *Señal de pulso arterial*
http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212005000200003

2.9 DETERMINACIÓN DE LA SEÑAL DE OXÍGENO (SPO2)

Para la determinación de la saturación de hemoglobina arterial con oxígeno (SpO_2), el oxímetro de pulso o pulsioxímetro usa la espectrofotometría basada en que la oxihemoglobina u hemoglobina oxigenada (HbO_2) y la desoxihemoglobina o hemoglobina reducida (Hb) absorben y transmiten determinadas longitudes de onda del espectro luminoso para la luz roja (640-660nm) y la luz infrarroja (910-940nm). La HbO_2 absorbe más la luz infrarroja y permite el paso de la luz roja; por el contrario, la Hb absorbe más la luz roja (R) y permite el paso de la luz infrarroja (IR). El ratio de la absorción de la luz R e IR mide el grado de oxigenación de la hemoglobina

El fotodiodo del sensor de oximetría mide tres niveles lumínicos diferentes: la luz roja, la luz infrarroja y también la luz ambiente. El principio que permite que el oxímetro sea un “oxímetro arterial” o “de pulso”, es que se basa en los valores medidos al momento de cada pulsación de la sangre arterial, ya que se supone que sólo la sangre arterial pulsa en el lugar de la medición, lo que se denomina el componente arterial (AC) pulsátil.

La luz absorbida cambia al variar la cantidad de sangre en el lecho tisular y al modificarse las cantidades relativas de HbO_2 y Hb . La medición de los cambios en la absorción de la luz permite estimar la SpO_2 arterial y la frecuencia cardiaca. Además, existe el componente estático (DC), formado por los tejidos, el hueso, los vasos sanguíneos, la piel y la sangre venosa. El máximo y el mínimo de la señal pulsátil detectada, tanto para la luz roja como para la luz infrarroja, corresponden, respectivamente, a las absorciones Adquisición de señales de oximetría de pulso para monitoreo del estado vascular 2-31 máximas y mínimas, las cuales, al ser divididas entre sí, normalizan la transmisión lumínica. Esto permite calcular el cociente R, que determina el nivel de SpO_2 del paciente, lo que se muestra en la ecuación (2.4). $R = \frac{RAB}{IAB}$ (2.4) siendo, RAB el cociente entre la absorción máxima y mínima del led rojo e IAB el cociente entre la absorción máxima y mínima del led infrarrojo. Esta ecuación hace necesario emplear una gráfica de regresión lineal para estimar la función existente entre el cociente R con la SpO_2 . Dicha gráfica de regresión, es determinada empíricamente con los valores de saturación obtenidos de

voluntarios humanos sometidos a diferentes pruebas, y luego se los contrasta con mediciones hechas por equipos comerciales.

2.10 LENGUAJE DE PROGRAMACION DE ALTO NIVEL

Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad con que los ejecutan las máquinas. Estos lenguajes permiten una máxima flexibilidad al programador a la hora de abstraerse o ser literal. Permiten un camino bidireccional entre el lenguaje máquina y una expresión casi oral entre la escritura del programa y su posterior compilación. Por lo general suelen estar orientados a objetos, a eventos, o a funciones, pudiendo estos combinarse. Asimismo, pueden ser compilados o interpretados. Algunos ejemplos son: java, visual basic.net, Python.¹³

2.11 VISUAL BASIC.NET

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es retro compatible con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.¹⁴

2.11.1 DIFERENCIAS DE VISUAL BASIC

Las diferencias entre VB y VB.NET son profundas, sobre todo en cuanto a metodología de programación y bibliotecas, pero ambos lenguajes siguen manteniendo un gran parecido, cosa que facilita notablemente el paso de VB a VB.NET.

¹³ Wikipedia, la enciclopedia libre, (2021). *Lenguaje de alto nivel*. Recuperado de https://es.wikipedia.org/Lenguaje_de_alto_nivel/

¹⁴ Wikipedia, la enciclopedia libre, (2021). *Visual Basic.NET*. Recuperado de https://es.wikipedia.org/wiki/Visual_Basic_.NET

2.11.2 TIPOS DE DATOS

Todos los lenguajes de programación que cumplen las normas de .NET tienen muchas cosas en común, una de ellas es el conjunto de tipos de datos. Hay que destacar que estos tipos de datos están implementados como clases, de manera que una variable declarada de un tipo determinado, tendrá la capacidad de usar tanto los métodos como las propiedades que pertenezcan a la clase del tipo de dato.¹⁵

En la siguiente tabla se muestra una relación de los tipos de datos de .NET Framework y su correspondencia en VB.NET y C#.

Nombre de la clase	Tipo de dato en VB.NET	Tipo de dato en C#	Descripción
Byte	Byte	Byte	Entero sin signo de 8 bit.
Sbyte	Sbyte (No nativo)	sbyte	Entero sin signo de 8bit (Tipo no acorde con el CLS)
Int16	Short	short	Entero con signo de 16 bit.
Int32	Integer	int	Entero con signo de 32 bit.
Int64	Long	long	Entero con signo de 64 bit.
UInt16	UInt16 (No nativo)	ushort	Entero sin signo de 16 bit. (Tipo no acorde con el CLS)
UInt32	UInt32 (No nativo)	uint	Entero sin signo de 32 bit. (Tipo no acorde con el CLS)
UInt64	UInt64 (No nativo)	ulong	Entero sin signo de 64 bit. (Tipo no acorde con el CLS)
Single	Single	float	Numero con coma flotante de precisión simple, de 32 bit.
Double	Double	double	Numero con coma flotante de precisión doble, de 64 bit.
Boolean	Boolean	bool	Valor logico
Char	Char	char	Carácter unicode de 16 bit.
Decimal	Decimal	decimal	Valor decimal de 96 bit.

¹⁵ Desarrollo Web. (2022). *Tipos de datos en .NET*. Recuperado de <https://desarrolloweb.com/articulos/1388.php>

IntPtr	IntPtr (No nativo)	--	Entero con signo cuyo tamaño depende de la plataforma: 32 bit en plataformas de 32 bit y 64 bit en plataformas de 64 bit. (Tipo no acorde con el CLS)
UIntPtr	UIntPtr (No nativo)	--	Entero sin signo cuyo tamaño depende de la plataforma: 32 bit en plataformas de 32 bit y 64 bit en plataformas de 64 bit. (Tipo no acorde con el CLS)
String	String	string	Cadena de caracteres.

2.11.3 DATAGRIDVIEW CON VISUAL BASIC .NET

Un DataGridView es un control en visual el cual te permite mostrar información al usuario en forma de una tabla.

	ID	Nombre	Apellido	Correo
▶	1	Hugo	Ortega	hugo.ortega@mail.com
	2	Alexa	Martinez	alexa@mail.com
	3	Julio	Mendieta	juliomen@mail.com
	4	Ernesto	Aguilera	ernesto-aguilera@mail.com
	5	Ulises	Quintana	uli-quintana@mail.com
	6	Fernando	Rodriguez	ferrod@mail.com
*				

Figura 10: Tabla informativa aplicando datagridview.

Fuente: <https://envb.net/datagridview/>

Un DataGridView te brinda la posibilidad de obtener la información que deseas presentar al usuario desde una base de datos o incluso, desde un servicio web.

Por otro lado, también te permite agregar manualmente los datos que deseas sin necesidad de conectarte a una fuente de datos como puede ser una base de datos.

Una de las ventajas es que te permite administrar, de manera práctica y sencilla, la información de la fuente de datos directamente desde el control.¹⁶

De esta manera podrías:

¹⁶ DataGridView. (2021). *Que es y para qué sirve un DataGridView*.
<https://envb.net/datagridview/>

Recuperado de

- Agregar datos
- Modificar datos
- Eliminar datos

Puede resultar muy práctico y sencillo manejar la información con este control.

Otra de las ventajas es que puedes personalizar a tu gusto y necesidad la apariencia del control estableciendo, por ejemplo:

- Los colores de fila
- El color de fondo del control
- El alto/ancho de las filas y columnas
- La alineación del texto
- La fuente
- Entre otro

2.11.4 CONDICIONALES Y BUCLES

If: Condicional que se ejecuta en un bloque interno siempre y cuando se cumpla una condición.

```
If condición1 Then
  Instrucción 1
Elseif condicion2 Then
  Instrucción 2
Else
  Otra Instrucción
End if
```

For: Es un bucle que al igual que if tiene una condición, pero además tiene un incremento o decremento.

```
For variable=inicio To fin Step paso
  Instrucciones
Next variable
```

While: Es un bucle que se cumple mientras la condición sea verdadera, es importante considerar que se pueden producir bucles infinitos.

```
While condición
```

```
Instrucciones End  
while
```

2.11.5 PROCEDIMIENTOS Y FUNCIONES

Los procedimientos ejecutan una serie de códigos sin devolver un resultado, en cambio las funciones ejecutan una serie de códigos y devuelven un resultado al código que la llamó.

```
Private Sub Procedimiento1()  
    Instrucciones  
End sub  
  
Private Function Funcion1(Parámetros)  
    Instrucciones  
    Return valor End  
Function
```

2.11.6 FORMULARIO Y CUADRO DE HERRAMIENTAS

El formulario es la parte más importante dentro de una aplicación GUI, ya que es la interfaz gráfica entre el usuario y la máquina, por lo que un formulario debe ser claro entendible y ordenado en cuanto a los controles que posee; como ser botones, etiquetas, barras de progreso y todos los controles disponibles en el cuadro de herramientas o controles creados por el usuario.

2.11.7 CONTROLES DE USUARIO

Los lenguajes .NET tienen la posibilidad de crear controles personalizados pudiendo tener estos controles básicos (botones, cuadros de texto, etiquetas, etc.) incorporados para el nuevo control. Empresas como *National Instruments*, *Bunifu* o *DevComponents* han realizado varios controles personalizados como ser: Botones con diferentes estilos o colores, barras de progreso circulares, controles de medición como ser los denominados «*Gauge*». Estos controles dan una apariencia más profesional a una aplicación de escritorio teniendo un estilo visual más atractivo y amigable.

2.12 BUNIFU FRAMEWORK

Los controles de interfaz de usuario de *Bunifu* son herramientas impulsadas por DLL para ayudarlo a crear impresionantes interfaces de aplicaciones de escritorio. Garantiza una

excelente experiencia de usuario en sus aplicaciones y reduce el tiempo de desarrollo para los desarrolladores de software de Microsoft Visual Studio .NET.

Actualmente los lenguajes C# y VB.NET y el producto se ofrece como una DLL importada a su entorno de Visual Studio.

Las DLL se instalan en su entorno .NET y se basan en WinForms. Permite a los desarrolladores arrastrar y soltar, por lo tanto, una codificación más rápida.¹⁷

Se dirigen a:

- 1) Desarrolladores de software individuales
- 2) Empresas de software
- 3) Empresas industriales que requieren automatización

2.13 SOCKETS EN VB.NET

Los Socket proporciona un amplio conjunto de métodos y propiedades para las comunicaciones de red. La clase Socket le permite realizar transferencias de datos sincrónicas y asincrónicas utilizando cualquiera de los protocolos de comunicación enumerados en la enumeración ProtocolType .

La clase Socket sigue el patrón de nomenclatura de .NET Framework para métodos asincrónicos. Por ejemplo, el método Receive síncrono corresponde a los métodos BeginReceive y EndReceive asíncronos .¹⁸

Si su aplicación solo requiere un subproceso durante la ejecución, use los siguientes métodos, que están diseñados para el modo de operación síncrona.

- Si está utilizando un protocolo orientado a la conexión como TCP, su servidor puede escuchar conexiones utilizando el método Listen . El método Accept procesa cualquier solicitud de conexión entrante y devuelve un Socket que puede usar para

¹⁷ Microsoft AppSource, (2022). *Bunifu Framework-Visual Studio* Recuperado de <https://appsource.microsoft.com/eses/product/webapps/bunifutechnologiesltd.bunifudevtools?tab=overvie>

¹⁸ Microsoft Documentos, (2022). *Familia de soluciones tipo sockets* Recuperado de <https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.socket?view=net-6.0#constructors>

comunicar datos con el host remoto. Utilice este Socket devuelto para llamar al método Send o Receive . Llame al método Bind antes de llamar al método Listen si desea especificar la dirección IP local y el número de puerto. Utilice un número de puerto de cero si desea que el proveedor de servicios subyacente le asigne un puerto gratuito. Si desea conectarse a un host de escucha, llame a Connect método. Para comunicar datos, llame al método Enviar o Recibir .

- Si está utilizando un protocolo sin conexión como UDP, no necesita escuchar las conexiones en absoluto. Llame al método ReceiveFrom para aceptar cualquier datagrama entrante. Utilice el método SendTo para enviar datagramas a un host remoto.

Para procesar las comunicaciones mediante subprocesos separados durante la ejecución, utilice los siguientes métodos, que están diseñados para el modo de funcionamiento asíncrono.

- Si usa un protocolo orientado a la conexión como TCP, use los métodos Socket , BeginConnect y EndConnect para conectarse con un host de escucha. Utilice los métodos BeginSend y EndSend o BeginReceive y EndReceive para comunicar datos de forma asíncrona. Las solicitudes de conexión entrantes se pueden procesar mediante BeginAccept y EndAccept .
- Si utiliza un protocolo sin conexión como UDP, puede utilizar BeginSendTo y EndSendTo para enviar datagramas y BeginReceiveFrom y EndReceiveFrom para recibir datagramas.

Si realiza varias operaciones asíncronas en un socket, no necesariamente se completan en el orden en que se iniciaron.

Cuando haya terminado de enviar y recibir datos, utilice el método de apagado para desactivar el Socket . Después de llamar a Shutdown , llame al método Close para liberar todos los recursos asociados con el Socket .

2.14 ARDUINO

Arduino es una compañía de desarrollo de *software* y *hardware* libre, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.¹⁹

2.14.1 CARACTERÍSTICAS DEL IDE DE ARDUINO

El entorno de desarrollo integrado también llamado IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes. Un IDE consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en su memoria flash.

Los programas escritos usando el software Arduino (IDE) se llaman bocetos. Estos bocetos están escritos en el editor de texto y se guardan con la extensión de “archivo.ino”.

¹⁹ Wikipedia, la enciclopedia libre, (2021).
<https://es.wikipedia.org/wiki/Arduino>

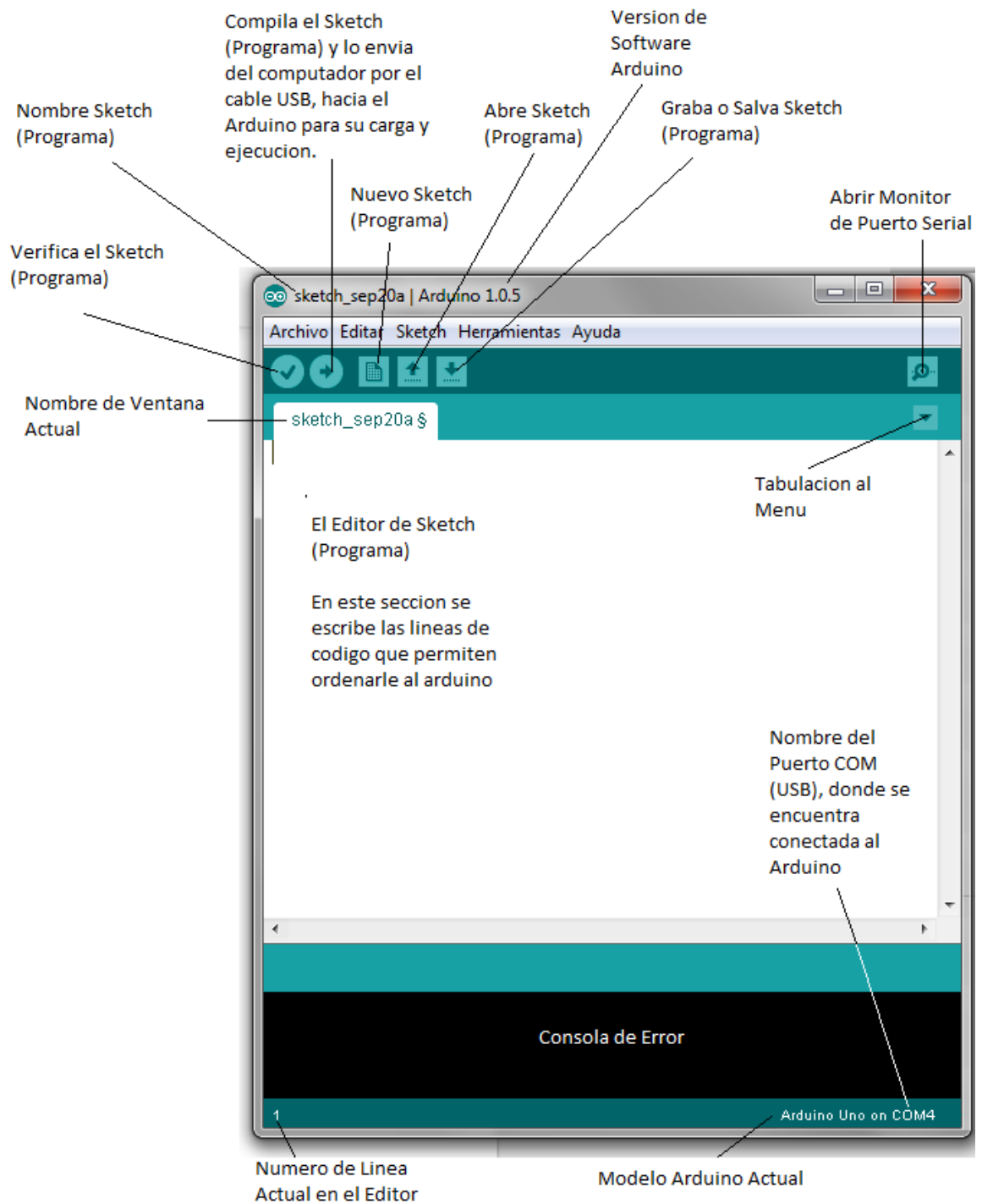


Figura 11: IDE de Arduino.

Fuente: <https://sites.google.com/site/xtalejandrovaldivia/p/proyecto-4>

El editor tiene funciones para cortar / pegar y para buscar / reemplazar texto. El área de mensajes brinda comentarios al guardar y exportar y también muestra errores. La consola muestra el texto generado por el software Arduino (IDE), incluidos los mensajes de error completos y otra información. La esquina inferior derecha de la ventana muestra la placa

configurada y el puerto serie. Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar bocetos, y abrir el monitor en serie como se muestra en la Figura 10.

Para conocer más a la estructura básica del lenguaje de programación de Arduino, la página de oficial nos provee de mucha documentación el cual nos puede ayudar a comprender como se programan en estas plataformas, su programación es simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones como se muestra en la Figura 10.

En el segmento `setup()` es la parte encargada de recoger la configuración inicial y el segmento `loop()` es la que contiene el programa que se ejecutará cíclicamente (de ahí el término loop –bucle-). Ambas funciones son necesarias para que el programa trabaje correctamente.²⁰

2.15 PROTOCOLO DE INTERNET VERSIÓN 4 IPV4

El Protocolo de Internet versión 4 (en inglés, *Internet Protocol version 4, IPv4*), es la cuarta versión del Internet Protocol (IP), un protocolo de interconexión de redes basados en Internet, y que fue la primera versión implementada en 1983 para la producción de ARPANET.

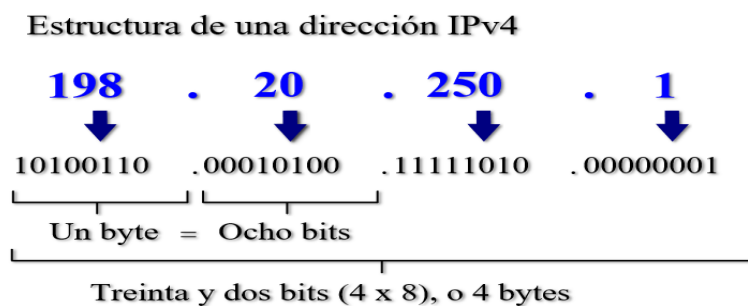


Figura 12: Detalle de Una Dirección IPV4.
Fuente: <https://www.alegsa.com.ar/Dic/ipv4.php>

²⁰ Aprendiendo Arduino Worldexpress, (2021).
<https://aprendiendoarduino.wordpress.com/2017/06/18/ide-arduino-y-configuracion/>

2.16 PROTOCOLO TCP

Las direcciones IPv4 se pueden escribir de forma que expresen un entero de 32 bits, aunque normalmente se escriben con decimales separados por puntos. A estos números decimales de 3 dígitos se les llama «octetos», porque en binario requieren de 8 dígitos (8 bits) para ser representados.²¹

TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

Con el uso del protocolo TCP, las aplicaciones pueden comunicarse de forma segura (gracias al sistema de acuse de recibo ACK)

Durante la comunicación usando el protocolo TCP, las dos máquinas deben establecer una conexión. La máquina emisora (la que solicita la conexión) se llama cliente, y la máquina receptora se llama servidor.²²

2.17 WIFI

WiFi es una tecnología que permite la conexión de equipos electrónicos a una red de manera inalámbrica.

Wi-Fi es una marca de la alianza Wi-Fi, la organización comercial que cumple con los estándares 802.11 relacionados con redes inalámbricas de área local.²³

La abreviación de WiFi significa *wireless fidelity* traducido como *fidelidad inalámbrica*.

2.18 SERVIDOR

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador

²¹ Wikipedia, la enciclopedia libre, (2020). *IPV4*. Recuperado de <https://es.wikipedia.org/wiki/IPv4>

²² CCM (2020). *Protocolo TCP*. Recuperado de <http://es.ccm.net/contents/281-protocolo-tcp#que-es-elprotocolo-tcp>

²³ Wikipedia, la enciclopedia libre, (2020). *WiFi*. Recuperado de <https://es.wikipedia.org/wiki/Wifi>

físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.²⁴

Un servidor sirve información a los equipos que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a datos y otra información del servidor.

Un servidor tiene la capacidad de trabajar de manera permanente, esto quiere decir 24 horas al día durante los 7 días de la semana.

2.19 RS-232

RS-232 (*Recommended Standard 232*, en español: «*Estándar Recomendado 232*»), también conocido como EIA/TIA RS-232C, es una interfaz que designa una norma para el intercambio de datos binarios serie entre un DTE (*Data Terminal Equipment*, «*Equipo Terminal de Datos*»), como por ejemplo una computadora, y un DCE (*Data Communication Equipment*, «*Equipo de Comunicación de Datos*»), por ejemplo, un módem.

En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos, como pueden ser computadoras. Evidentemente, en el caso de interconexión entre los mismos, se requerirá la conexión de un DTE con otro DTE. Para ello se utiliza una conexión entre los dos DTE sin usar módem, por ello se llama módem nulo (*null modem*).



Figura 13: Conector RS-232 (DE-9 Macho).
Fuente: <https://www.eltima.com/es/article/9-pin-serial-port.html>

²⁴ Más Adelante, (2020).
<https://www.masadelante.com/faqs/servidor>

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9, o popularmente mal denominados DB-9), más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie de la PC).

Las UART o U(S)ART (Transmisor y Receptor Asíncrono Universal) se diseñaron para convertir las señales que maneja la CPU y transmitir las al exterior. Las UART deben resolver problemas tales como la conversión de tensiones internas del DCE con respecto al DTE, gobernar las señales de control, y realizar la transformación desde el bus de datos de señales en paralelo a serie y viceversa. Debe ser robusta y deberá tolerar circuitos abiertos, cortocircuitos y escritura simultánea sobre un mismo pin, entre otras consideraciones. Es en la UART en donde se implementa la interfaz.

Generalmente, cuando se requiere conectar un microcontrolador (con señales típicamente entre 3,3 y 5 V) con un puerto RS-232 estándar, se utiliza un driver de línea, típicamente un MAX232 o compatible, el cual mediante dobladores de tensión positivos y negativos, permite obtener la señal bipolar (típicamente alrededor de +/- 6V) requerida por el estándar.

Para los propósitos de la RS-232 estándar, una conexión es definida por un cable desde un dispositivo al otro. Hay 25 conexiones en la especificación completa, pero es muy probable que se encuentren menos de la mitad de éstas en una interfaz determinada. La causa es simple, una interfaz full duplex puede obtenerse con solamente 3 cables.²⁵

2.20 MICROPROCESADOR Y MICROCONTROLADOR

2.20.1 MICROPROCESADOR

El microprocesador (o simplemente procesador) es el circuito integrado central más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el «cerebro» de un ordenador.

²⁵ Wikipedia, la enciclopedia libre, (2020).
<https://es.wikipedia.org/wiki/RS-232>

Es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de usuario; solo ejecuta instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar, dividir, las lógicas binarias y accesos a memoria.

Puede contener una o más unidades centrales de procesamiento (CPU) constituidas, esencialmente, por registros, una unidad de control, una unidad aritmética lógica (ALU) y una unidad de cálculo en coma flotante (conocida antiguamente como «coprocesador matemático»²⁶).

2.20.2 MICROCONTROLADOR

Un microcontrolador (abreviado μ C, UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.²⁷

2.21 ESP-32

ESP32 es una serie de sistemas de bajo costo y bajo consumo de energía en un microcontrolador de chip con Wi-Fi integrado y Bluetooth de modo dual. La serie ESP32 emplea un microprocesador Tensilica Xtensa LX6 en variaciones de doble núcleo y un solo núcleo e incluye interruptores de antena integrados, balun RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros y módulos de administración de energía. ESP32 es creado y desarrollado por Espressif Systems, una compañía china con sede en Shanghai, y TSMC lo fabrica utilizando su proceso de 40 nm. Es el sucesor del micro controlador ESP8266.²⁸

²⁶ Wikipedia, la enciclopedia libre, (2020). *Microprocesador*. Recuperado de <https://es.wikipedia.org/wiki/Microprocesador>

²⁷ Wikipedia, la enciclopedia libre, (2020). *Microcontrolador*. Recuperado de <https://es.wikipedia.org/wiki/Microcontrolador>

²⁸ Wikipedia, la enciclopedia libre, (2020). *ESP32*. Recuperado de <https://en.wikipedia.org/wiki/ESP32>

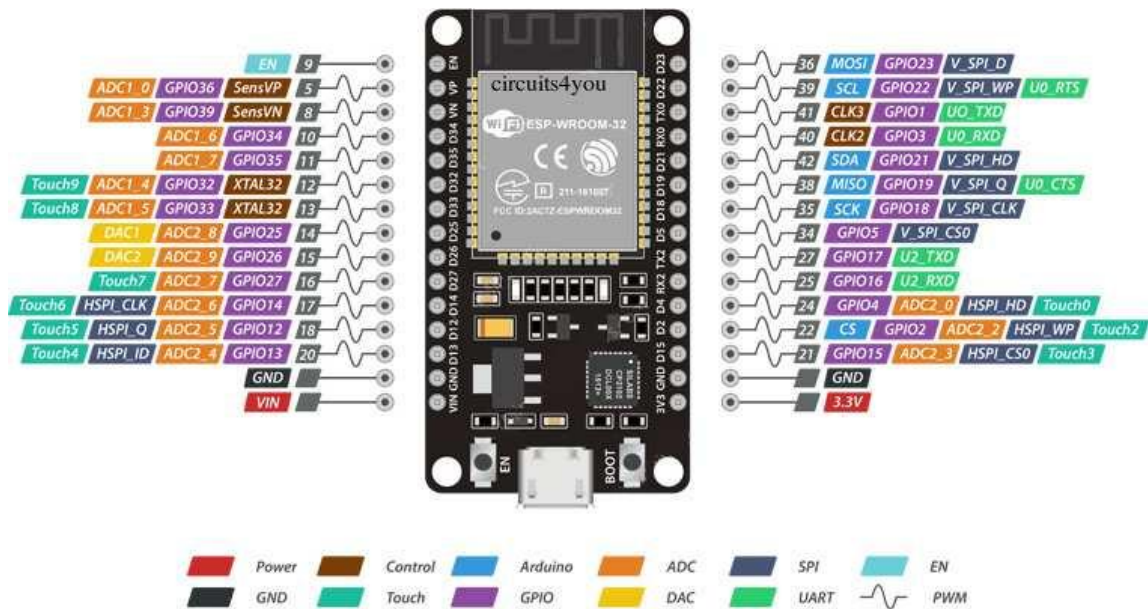


Figura 14: ESP32.

Fuente: <https://naylampmechatronics.com/espressif-esp/384-nodemcu-32-30-pin-esp32-wifi.html>

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- Placa: ESP32 DEVKIT V1 (Espressif)
- SoM: ESP-WROOM-32 (Espressif)
- SoC: ESP32 (ESP32-D0WDQ6)
- Frecuencia de Reloj: hasta 240Mhz
- Pines: 30
- Chip USB-Serial: CP2102
- Antena en PCB

2.21.1 CARACTERISTICAS DEL ESP32

Las características del ESP32 incluyen lo siguiente:

Procesadores:

- CPU: microprocesador LX6 de 32 bits de doble núcleo de Xtensa, que funciona a 160 o 240 MHz y tiene un rendimiento de hasta 600 DMIPS.

- Coprocesador de ultra baja potencia (ULP).

Memoria: 520 KB SRAM.

Conectividad inalámbrica:

- Wi-Fi: 802.11 b / g / n.
- Bluetooth: v4.2 BR / EDR y BLE.

Interfaces periféricas:

- ADC SAR de 12 bits hasta 18 canales.
- DAC de 2×8 bits.
- 10 sensores táctiles (GPIO de detección capacitiva).
- $4 \times$ SPI.
- $2 \times$ interfaces I²S.
- $2 \times$ interfaces I²C.
- $3 \times$ UART.
- Controlador de host SD / SDIO / CE-ATA / MMC / eMMC.
- Controlador esclavo SDIO / SPI.
- Interfaz Ethernet MAC con soporte de protocolo de tiempo de precisión DMA e IEEE 1588 dedicado.
- CAN bus 2.0.
- Mando a distancia por infrarrojos (TX / RX, hasta 8 canales).
- Motor PWM.
- LED PWM (hasta 16 canales).
- Sensor de efecto Hall.
- Preamplificador analógico de ultra baja potencia.

Seguridad:

- Todas las funciones de seguridad estándar IEEE 802.11 son compatibles, incluidas WPA, WPA / WPA2 y WAPI.
- Arranque seguro.

- Cifrado flash.
- OTP de 1024 bits, hasta 768 bits para clientes.
- Aceleración de hardware criptográfico: AES, SHA-2, RSA, criptografía de curva elíptica (ECC), generador de números aleatorios (RNG).

Administración de energía:

- Regulador interno de baja caída.
- Dominio de potencia individual para RTC.
- 5 μ A de corriente de sueño profundo.
- Despertar de la interrupción GPIO, temporizador, mediciones de ADC, interrupción capacitiva del sensor táctil.

2.22 SENSOR MAX30102

El increíble circuito integrado MAX30102 es un oxímetro y medidor de velocidad de biotransmisión de latidos del corazón. Integra un LED rojo y un LED infrarrojo, detectores fotoeléctricos, dispositivos ópticos y circuitos electrónicos de bajo ruido y de supresión de luz ambiental. Este circuito integrado puede ser alimentado con solo 1.8V y un voltaje diferente de 5V para la alimentación de los LEDs. Puede ser usado en los dedos, en el lóbulo de la oreja y la muñeca.

Es compatible con la interfaz I2C por lo que sus datos pueden ser transmitidos hacia algún Arduino o a cualquier microcontrolador que soporte I2C para hacer cálculos de velocidad de latidos y oxígeno en la sangre. En adición, el chip también puede apagar el módulo mediante software con una corriente standby de casi cero. La fuente de poder siempre se mantiene

2.22.1 CARACTERISTICAS DEL SENSOR MAX30102

Características principales:

- Tiene como método la fotodisolución que es la medida del pulso y la saturación de oxígeno es llevada a cabo usando tejido humano para causar diferentes transmitancias de la luz cuando las venas laten.

- Fuente de luz, una longitud de onda específica desde el diodo LED se selecciona para la oxihemoglobina (HbO₂) y la hemoglobina (Hb) en la sangre arterial.
- La transmitancia de luz, es convertida en una señal eléctrica, el cambio en el volumen de las pulsaciones arteriales causa que la transmitancia de la luz cambie. En este momento la luz reflejada por el tejido humano es recibida por el transductor fotoeléctrico, convertido en una señal eléctrica y amplificada para entregarla como salida.

Descripción de pines:

- Vin: Alimentación principal: 1.8-5V
- Pads 3-bits: Selecciona el nivel pull-up de el bus, dependiendo del voltaje maestro del pin, opcional 1.8V o 3.3V
- SCL: Clock conectado al bus I2C
- SDA: Datos conectada al bus I2C
- INT: Pin de interrupción
- RD: Terminal tierra del LED rojo del circuito integrado, generalmente no conectado.
- IRD: Terminal tierra del LED infrarrojo del circuito integrado, generalmente no conectado.
- GND: Tierra.

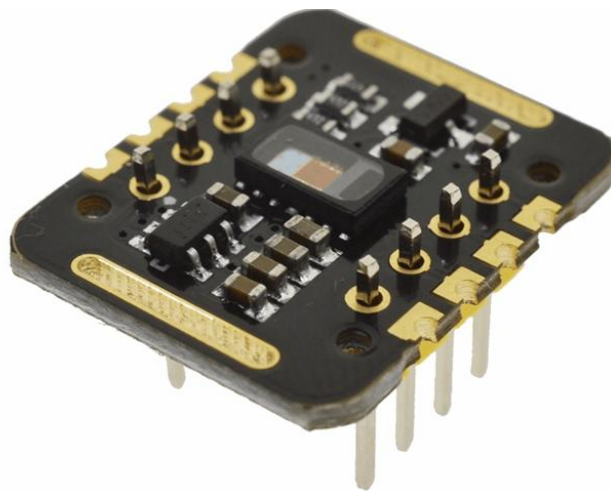


Figura 15: Sensor MAX30102

Fuente: <https://varpertechnologies.com/producto/sensor-de-de-pulsos-cardiacos-max30102-deteccion-de-oxigeno-en-la-sangre/?v=5bc574a47246>

CAPITULO III

MARCO PRÁCTICO

En esta fase se procederá a explicar el desarrollo y diseño del proyecto. Iniciando por el diseño, la elaboración del dispositivo tanto como la programación de software. En cuanto a este capítulo se tomará en cuenta los conceptos respecto a temas teóricos que vaya ligado al desarrollo que ayuden a comprender el funcionamiento del mismo.

3.1 DESCRIPCIÓN GENERAL DEL SISTEMA

Este es un sistema diseñado con el propósito de tener un dispositivo médico de bajo costo, con la capacidad de monitorear a un grupo masivo de pacientes esto con ayuda del uso de microcontroladores de la gama baja y alta de los microcontroladores, un sensor MAX30102 el cual proporciona datos de Oximetría de Pulso para una primera fase del proyecto. Comunicación por medio de una red inalámbrica, con transmisión y recepción de datos. Para la recepción de datos y visualización de los mismos se cuenta con una interfaz gráfica con la capacidad de mostrar el valor de saturación de oxígeno en la sangre (SPO2), el ritmo cardiaco (HR), dentro su característica posee un menú de configuración todo esto en la PC que permite realizar el seguimiento a los pacientes.

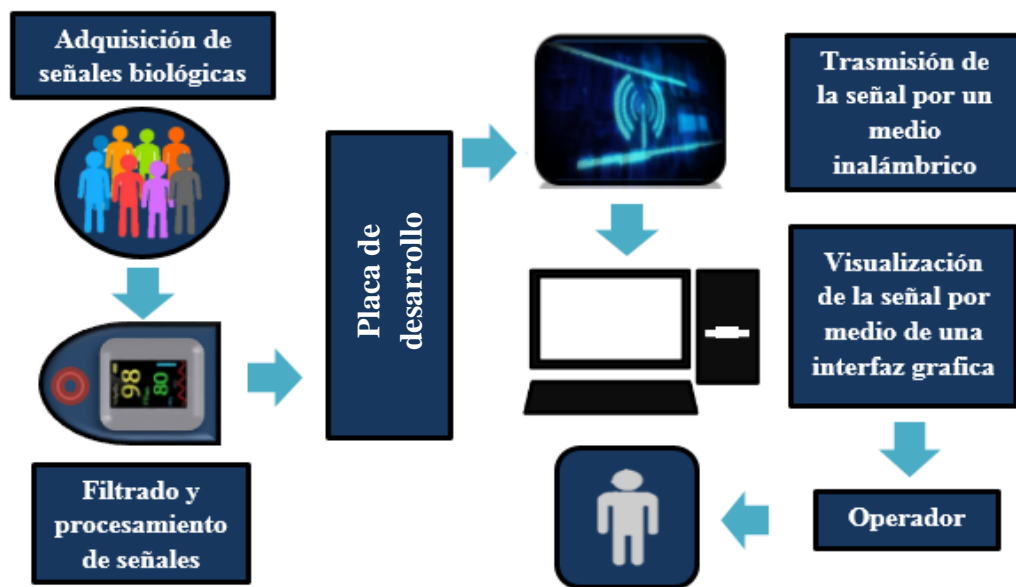


Figura 16: Diagrama básico del sistema de monitoreo.

Fuente: Diseño propio

El sistema cuenta con etapas para su monitoreo final, para la recepción y transmisión de datos es recomendable que el área local cuente con una red wifi.

El oxímetro está conectado al microcontrolador el cual envía datos de las señales cardiacas controlando niveles en tres estados:

- Primero estado, siendo este un estado pasivo (Nivel Bajo) que permanecerá inactivo dejando que las cosas ocurran sin su intervención.
- Segundo estado, (Nivel medio) tendrá un monitoreo con intervalos de tiempo y un estado activo.
- Tercer estado, (Nivel Alto) que estará siendo constantemente monitoreado enviando mensajes de atención inmediata.

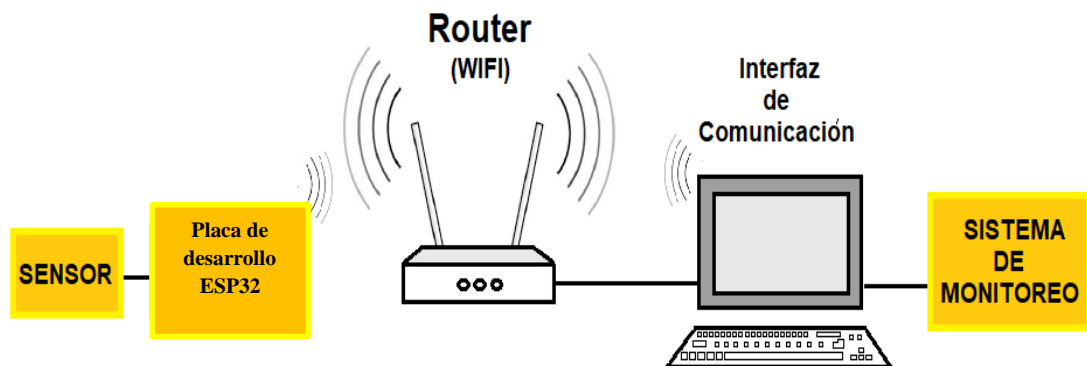


Figura 17: Diagrama general del dispositivo.

Fuente: Diseño propio

3.2 DESCRIPCIÓN DEL CIRCUITO DEL SISTEMA

La conformación principal de este dispositivo se fundamenta en siguiente esquema que se muestra en la Fig. 17, en la cual se muestra el desarrollo interno de este sistema principal que está compuesto a base de la utilización de la placa de desarrollo ESP32, sensor MAX30102 y un sistema de monitoreo.

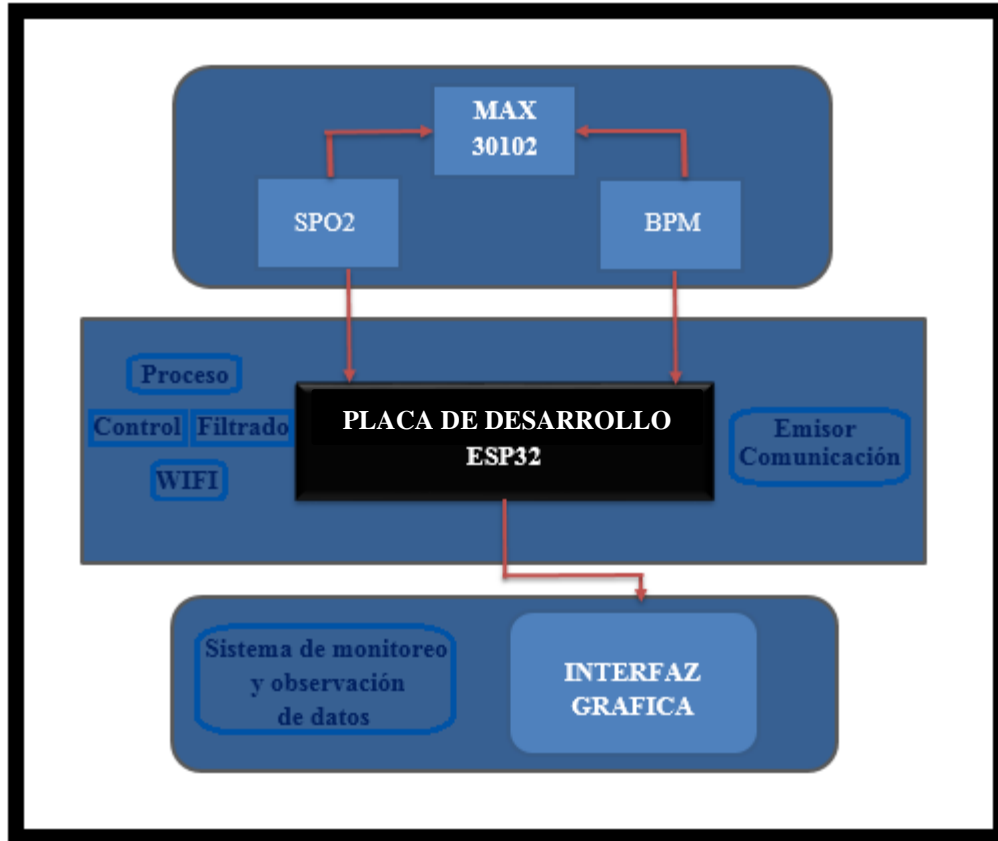


Figura 18: Diagrama de bloques del circuito principal.

Fuente: Diseño propio

Para el desarrollo del proyecto se empleó el funcionamiento del módulo ESP32, a pesar de ser un sistema en un chip de bajo costo y bajo consumo de energía presenta un WiFi de modo dual y Bluetooth integrado, además de la variedad de entornos de desarrollo de código abierto y las bibliotecas que lo hacen ideal para desarrolladores de todo tipo.

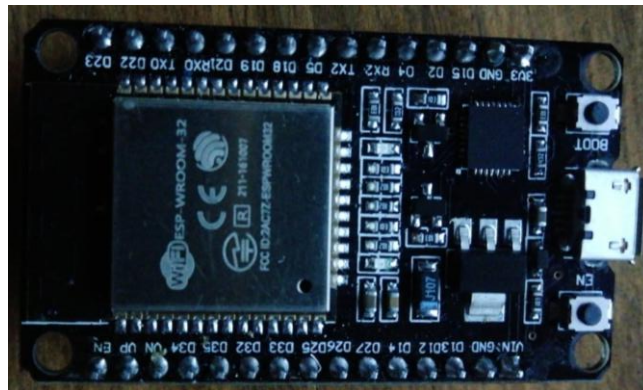


Figura 19: ESP Wroom 32 Dev. Kit V1.

Fuente: Imagen propia

El empleo de esta placa de desarrollo ESP32 en el proyecto permite por medio de sus pines pueda tener las lecturas del sensor MAX30102 tanto del SPO2 y BPM, permitiendo así una comunicación como la transmisión de datos, en este punto del proyecto se ha logrado cumplir uno de los objetivos trazados el cual era el medio de trasmisión de datos e interpretación de señales a distancia.

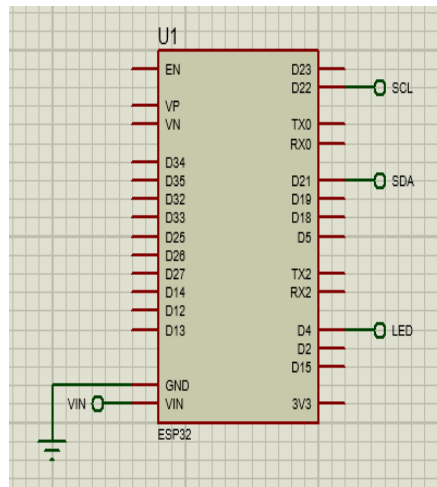


Figura 20: ESP32 diseñado en Proteus.

Fuente: Imagen correspondiente al esquema del circuito

3.3 PASOS A SEGUIR PARA TRABAJAR CON LA PLACA DE DESARROLLO ESP32

Para trabajar con la placa de desarrollo ESP32 se debe seguir una serie de pasos, inicialmente se debe tener instalado el IDE de Arduino.



Figura 21: IDE de Arduino en escritorio.

Fuente: Captura de pantalla

Una vez se ejecute el IDE de Arduino lo primero que se debe hacer es ingresar a la barra de opciones y hacer clic en “Archivo =>Preferencias” posterior a ello se nos aparecerá una ventana donde encontraremos el gestor de tarjetas adiciones.

Es justamente ahí donde se pegará una URL que previamente fue obtenida por medio de una página la cual otorga los links necesarios para poder descargarlos e instalarlos en distintos desarrolladores integrados para el caso se usara el siguiente URL: (https://dl.espressif.com/dl/package_esp32_index.json).



Figura 22: Frameworks de desarrollo oficial para el ESP32.

Fuente: Captura de pantalla

Esto se lo encuentra en su página oficial ESPRESSIF, una vez obtenida la URL que está disponible para todos los sistemas operativos disponibles una vez iniciado Arduino y después de haberse dirigido a la barra de herramientas “Archivo” seleccionar la opción de “Preferencias”.

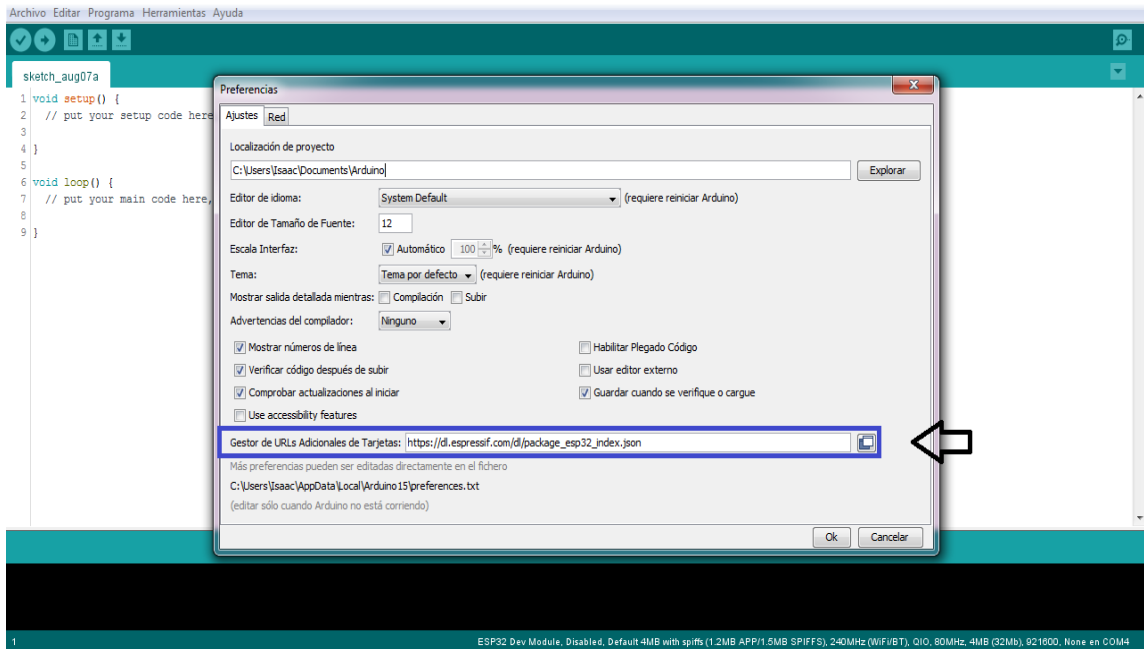


Figura 23: Gestor de URLs Adicionales de Tarjetas del IDE de Arduino.

Fuente: Captura de pantalla

Ingresar uno de los enlaces de lanzamiento en el campo URL del recuadro gestor de “URLs adicionales de tarjeta”.

Posterior a ello se debe ingresar Herramientas => Placa => Gestor de tarjetas una vez realizado este procedimiento se abrirá una ventana de Menú de tablero ahí se debe de instalar la plataforma ESP32.

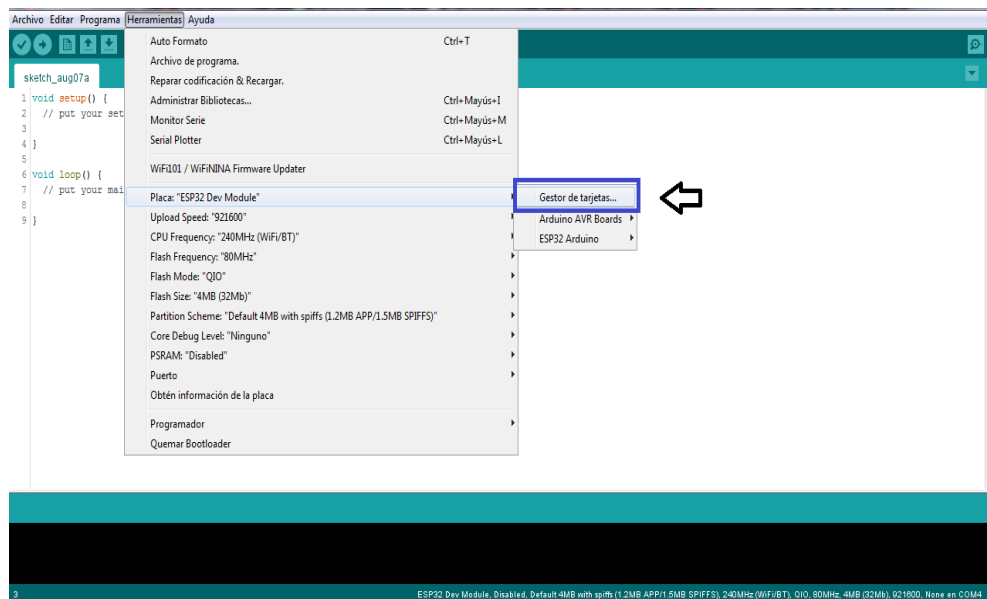


Figura 24: Gestor de tarjeta del IDE de Arduino.

Fuente: Captura de pantalla

Es recomendable que se instale en una de sus más recientes versiones.

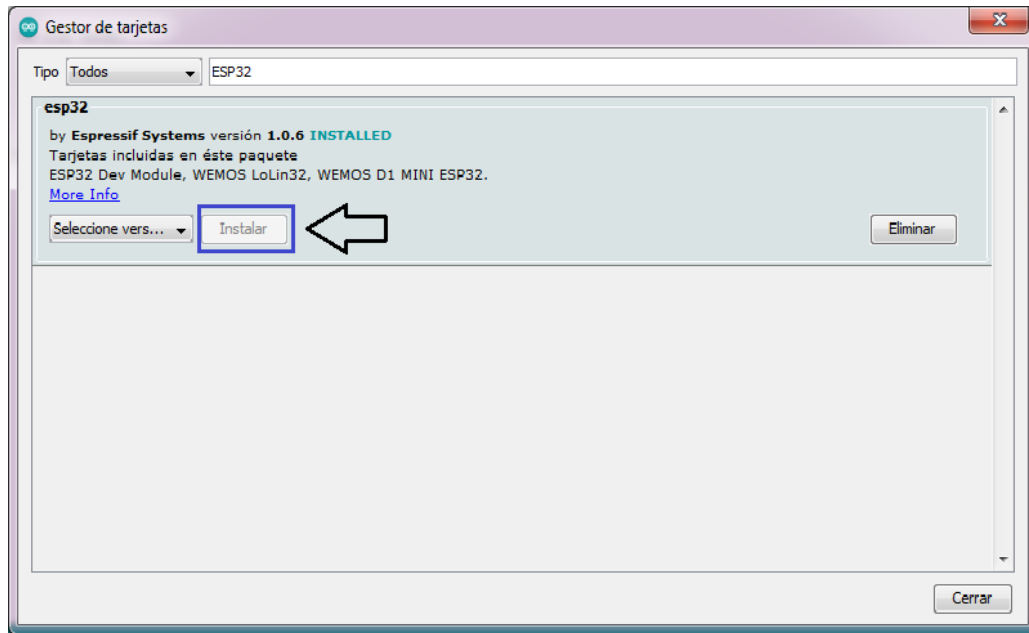


Figura 25: Tarjetas para el módulo ESP32.

Fuente: Captura de pantalla

Para verificar si se instaló correctamente los controladores se debe reiniciar el IDE de Arduino y seleccionar Herramientas => Placas y seleccionar la tarjeta ESP32 Dev module.

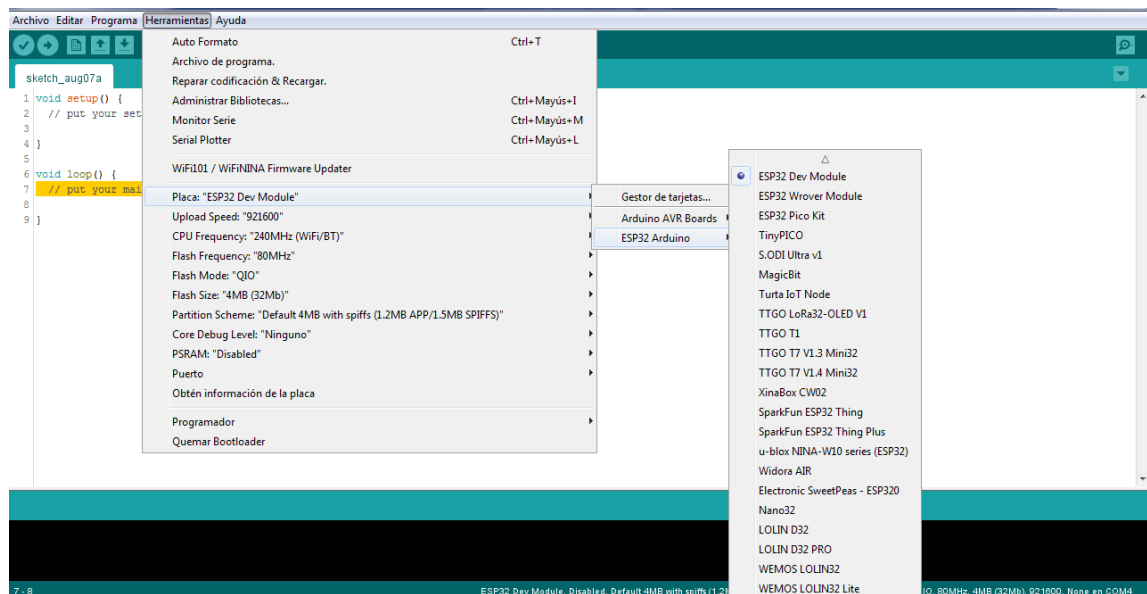
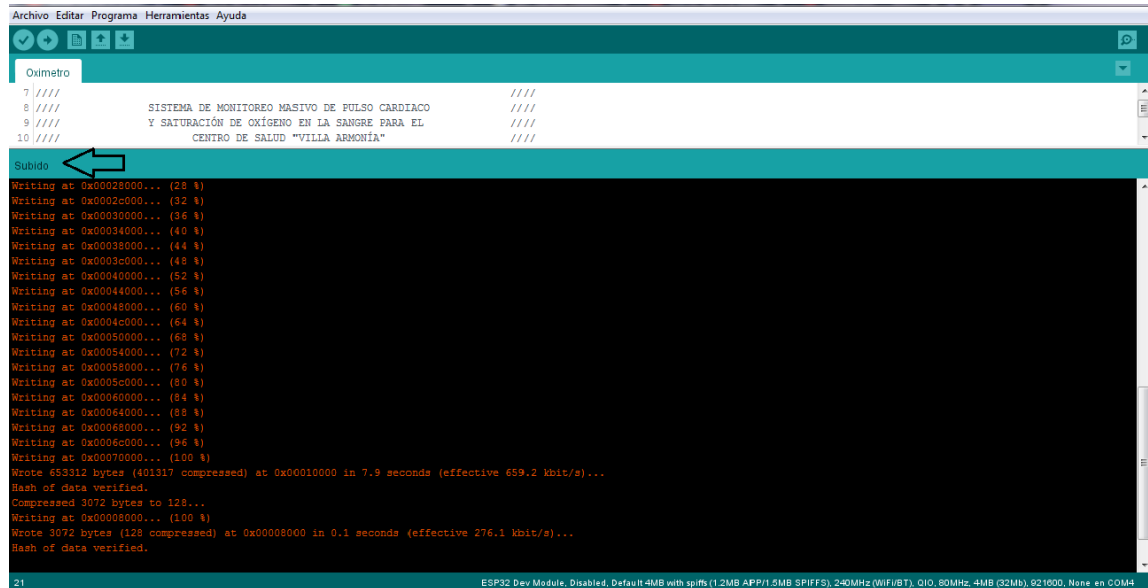


Figura 26: Selección del módulo ESP32.

Fuente: Captura de pantalla

Una vez terminado el proceso de reconocimiento de nuestra tarjeta ESP32 en un nuevo sketch se desarrollará el código para el funcionamiento de nuestro proyecto, si se instaló de forma correcta se compilará y se grabará el programa para nuestro proyecto de forma correcta como se muestra en la figura 26.



```
Archivo Editar Programa Herramientas Ayuda
Oxímetro
7 //
8 // SISTEMA DE MONITOREO MASIVO DE PULSO CARDIACO //
9 // Y SATURACIÓN DE OXÍGENO EN LA SANGRE PARA EL //
10 // CENTRO DE SALUD "VILLA ARMONÍA" //

Subido
Writing at 0x00028000... (28 %)
Writing at 0x0002c000... (32 %)
Writing at 0x00030000... (36 %)
Writing at 0x00034000... (40 %)
Writing at 0x00038000... (44 %)
Writing at 0x0003c000... (48 %)
Writing at 0x00040000... (52 %)
Writing at 0x00044000... (56 %)
Writing at 0x00048000... (60 %)
Writing at 0x0004c000... (64 %)
Writing at 0x00050000... (68 %)
Writing at 0x00054000... (72 %)
Writing at 0x00058000... (76 %)
Writing at 0x0005c000... (80 %)
Writing at 0x00060000... (84 %)
Writing at 0x00064000... (88 %)
Writing at 0x00068000... (92 %)
Writing at 0x0006c000... (96 %)
Writing at 0x00070000... (100 %)
Wrote 653312 bytes (401317 compressed) at 0x00010000 in 7.9 seconds (effective 659.2 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.1 seconds (effective 276.1 kbit/s)...
Hash of data verified.
```

Figura 27: Compilación y grabado.

Fuente: Captura de pantalla

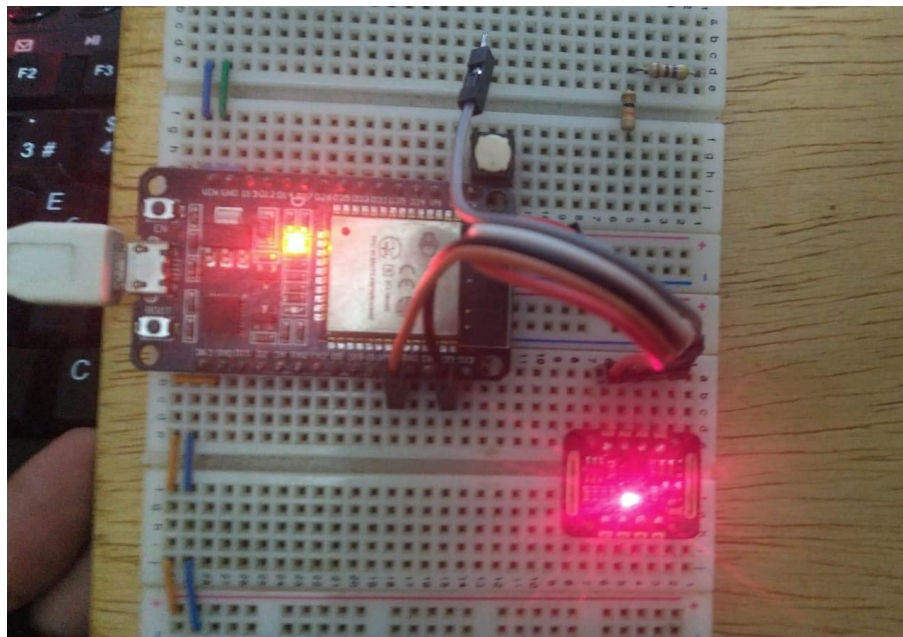


Figura 28: Verificado y subido al ESP32 exitosamente.

Fuente: Imagen Propia

3.4 INSTALACIÓN DE LIBRERÍAS NECESARIAS EN ARDUINO IDE

Las implementaciones de las librerías nos permitirán manejar y acceder a sus funciones de manera correcta del sensor MAX30102 para ello antes se debe seguir con los siguientes pasos:

Una vez en ejecución el IDE de Arduino se debe dirigir a la barra de opciones y seleccionar “Programa=>Incluir Librerías=>Administrar Bibliotecas”

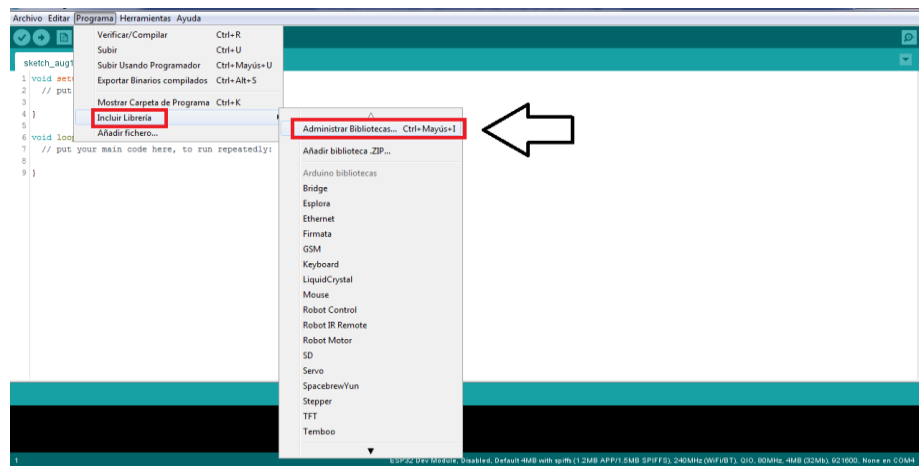


Figura 29: Incluir bibliotecas al IDE de Arduino

Fuente: Captura de pantalla

Aparecerá la siguiente ventana.

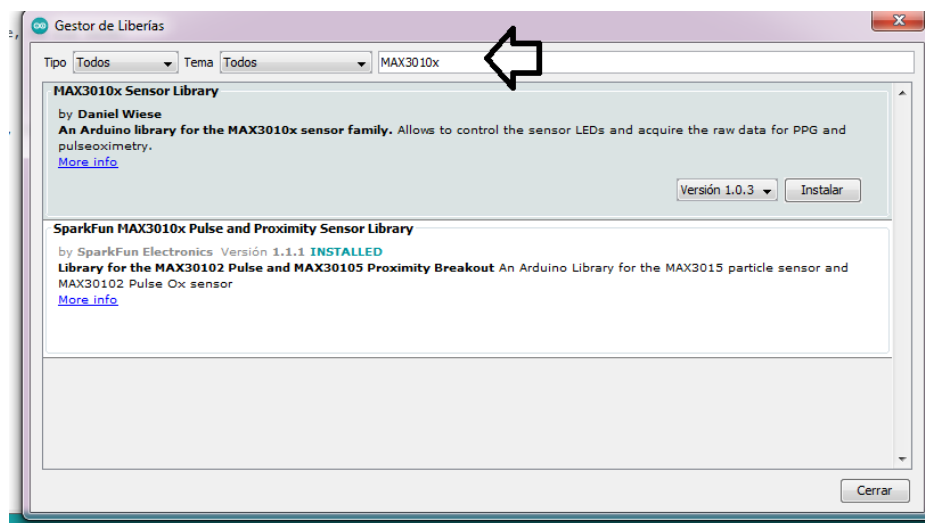


Figura 30: Gestor de librerías del IDE de Arduino.

Fuente: Captura de pantalla

Buscar la biblioteca MAX3010X e instalar la versión más reciente, una vez instalado la biblioteca se debe reiniciar el IDE de Arduino.

```
#include <MAX30105.h> //Incluye la libreria para el sensor MAX30102 ////  
#include <spo2_algorithm.h> //Incluye la libreria que obtiene valor de SP02 ////
```

Es así como podremos contar con las bibliotecas necesarias para el correcto funcionamiento de nuestro proyecto.

3.5 DISEÑO DEL CIRCUITO ESQUEMÁTICO E IMPRESO EN PROTEUS

Como se observa en la imagen inferior el diseño es bastante pequeño ya que no usa muchos componentes, lo que abaratará los costos de producción.

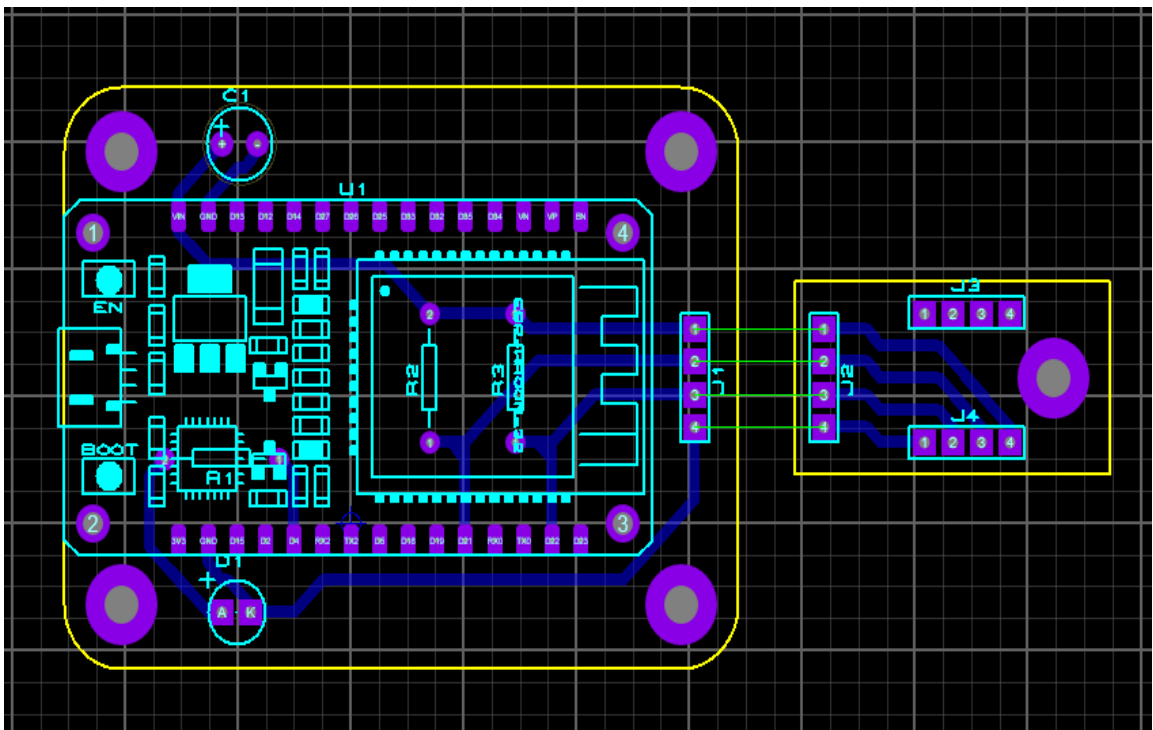


Figura 31: Simulación del proyecto.

Fuente: Captura de pantalla

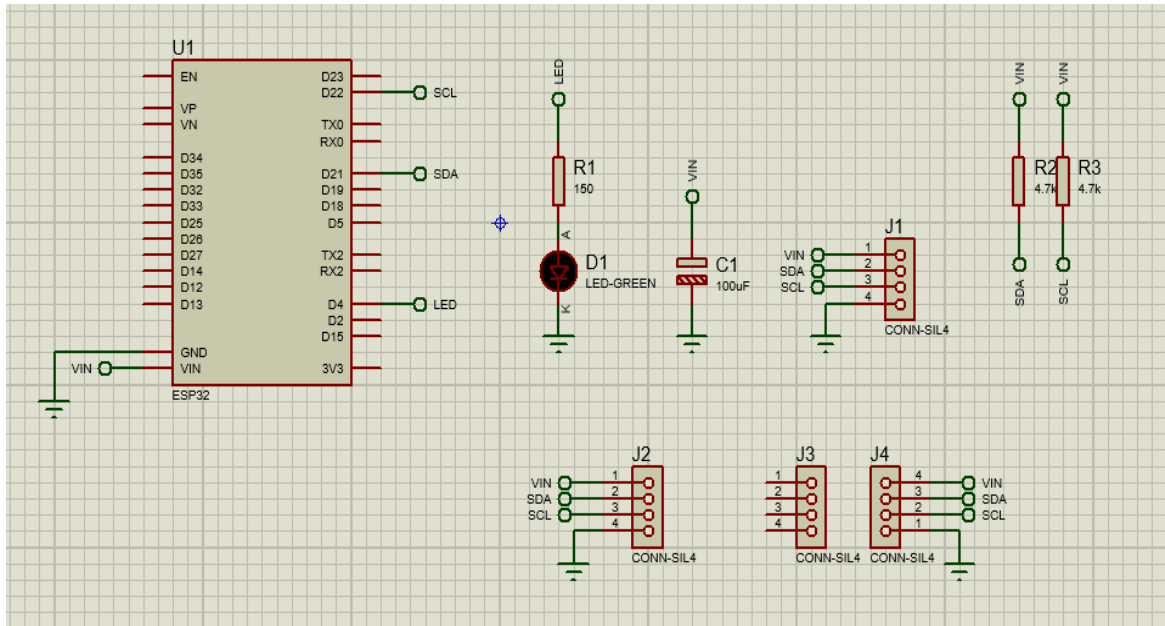


Figura 32: Esquema del proyecto.

Fuente: Imagen correspondiente al esquema del circuito

3.6 INTERFAZ DE PC PARA EL PROGRAMA DE MONITOREO

Se desarrolló un programa que es capaz de monitorear a los pacientes detectando las señales del oxímetro.

Al programa se le da el nombre de Samaña-Sunqu el cual proviene de dos idiomas originarios de Bolivia.

Samaña que viene del aimara y tiene como significado:

- Respiración
- Aliento
- Oxígeno
- Descanso

Sunqu que viene del quechua y tiene como significado:

- Corazón

Es por el significado traducido al español de ambas se lo denomina “Oxígeno del corazón”.

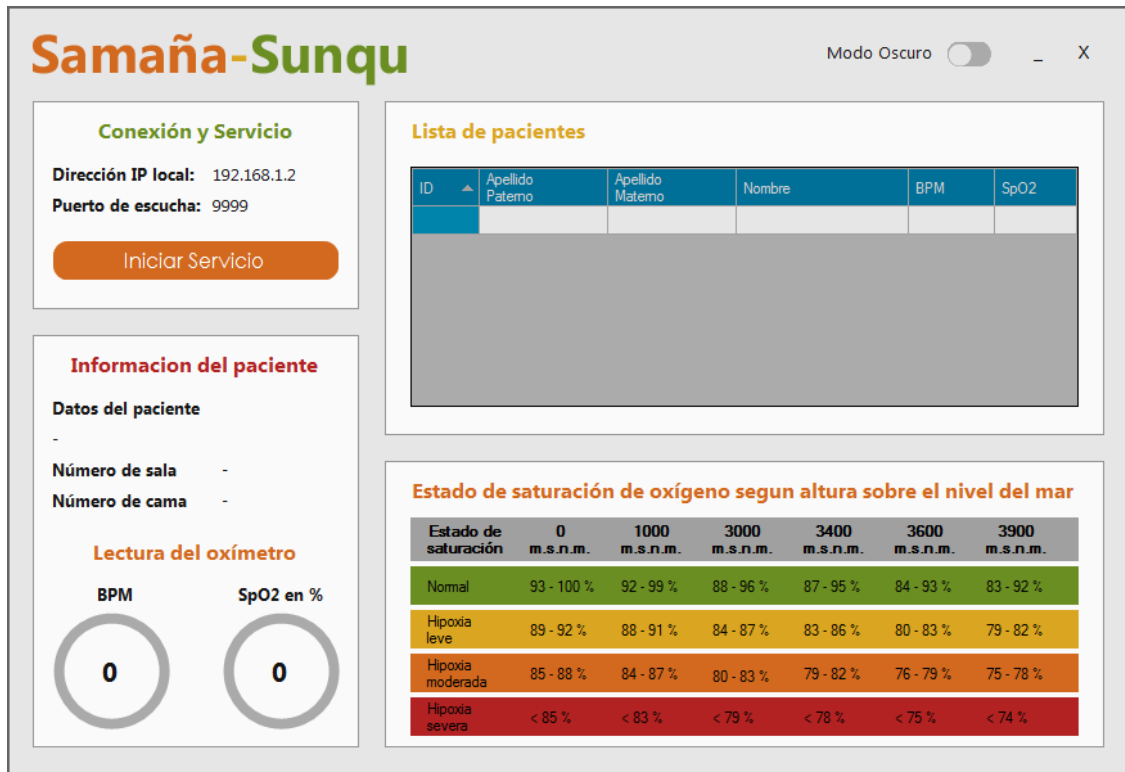


Figura 33: PC programa.

Fuente: Captura de Pantalla

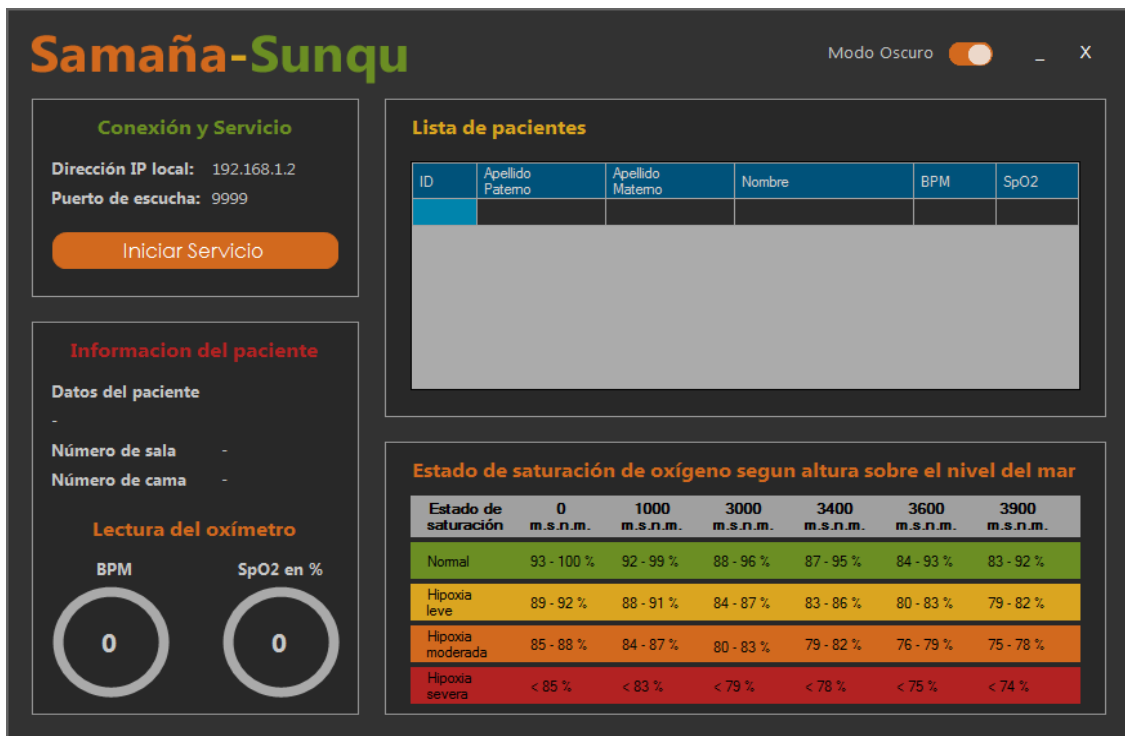


Figura 34: PC programa en modo oscuro.

Fuente: Captura de pantalla

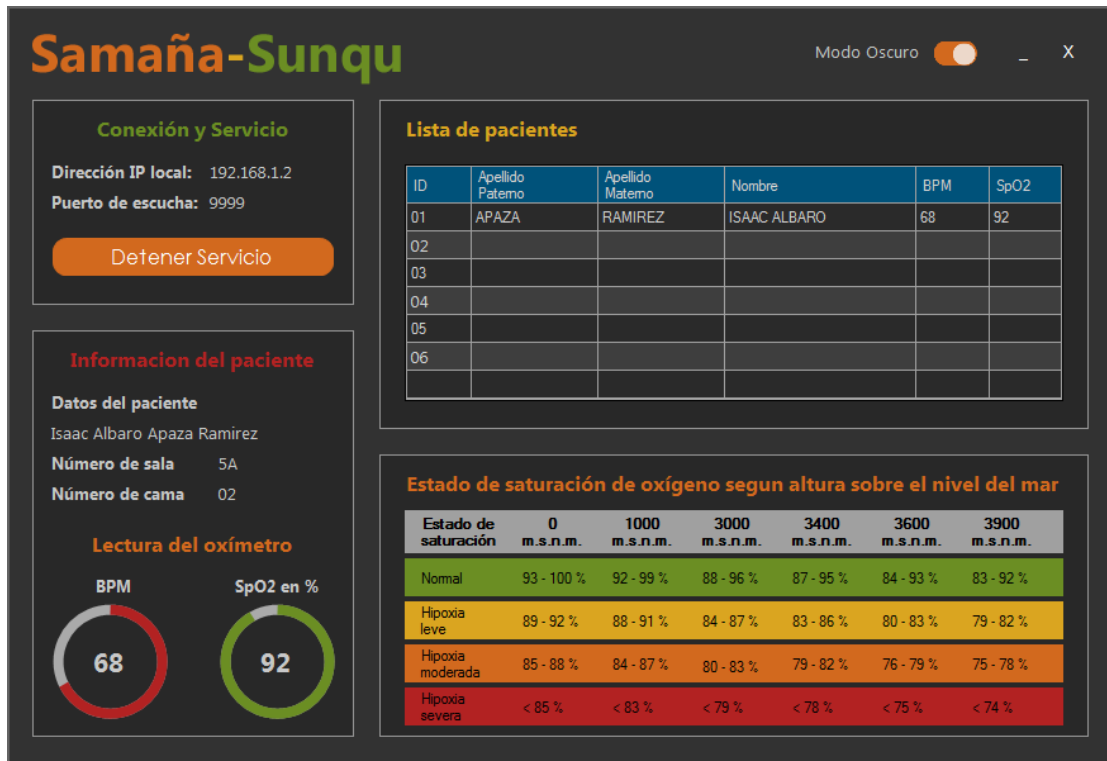


Figura 35: Recepción de datos en el PC programa modo oscuro.

Fuente: Captura de pantalla

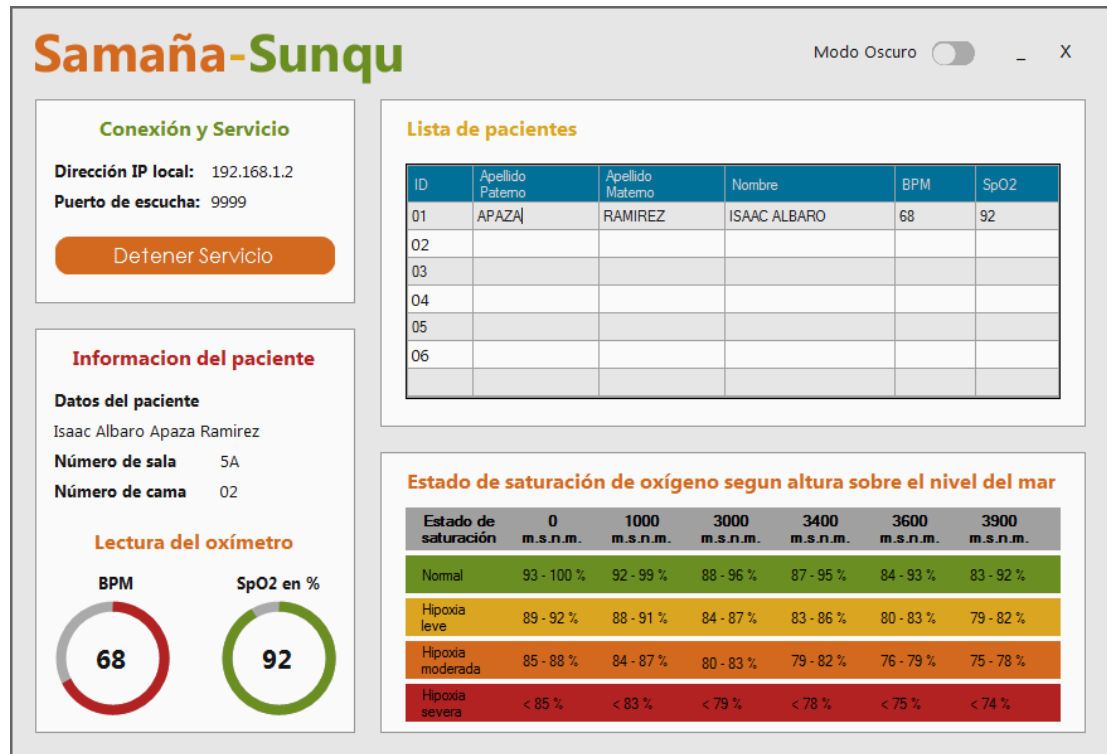
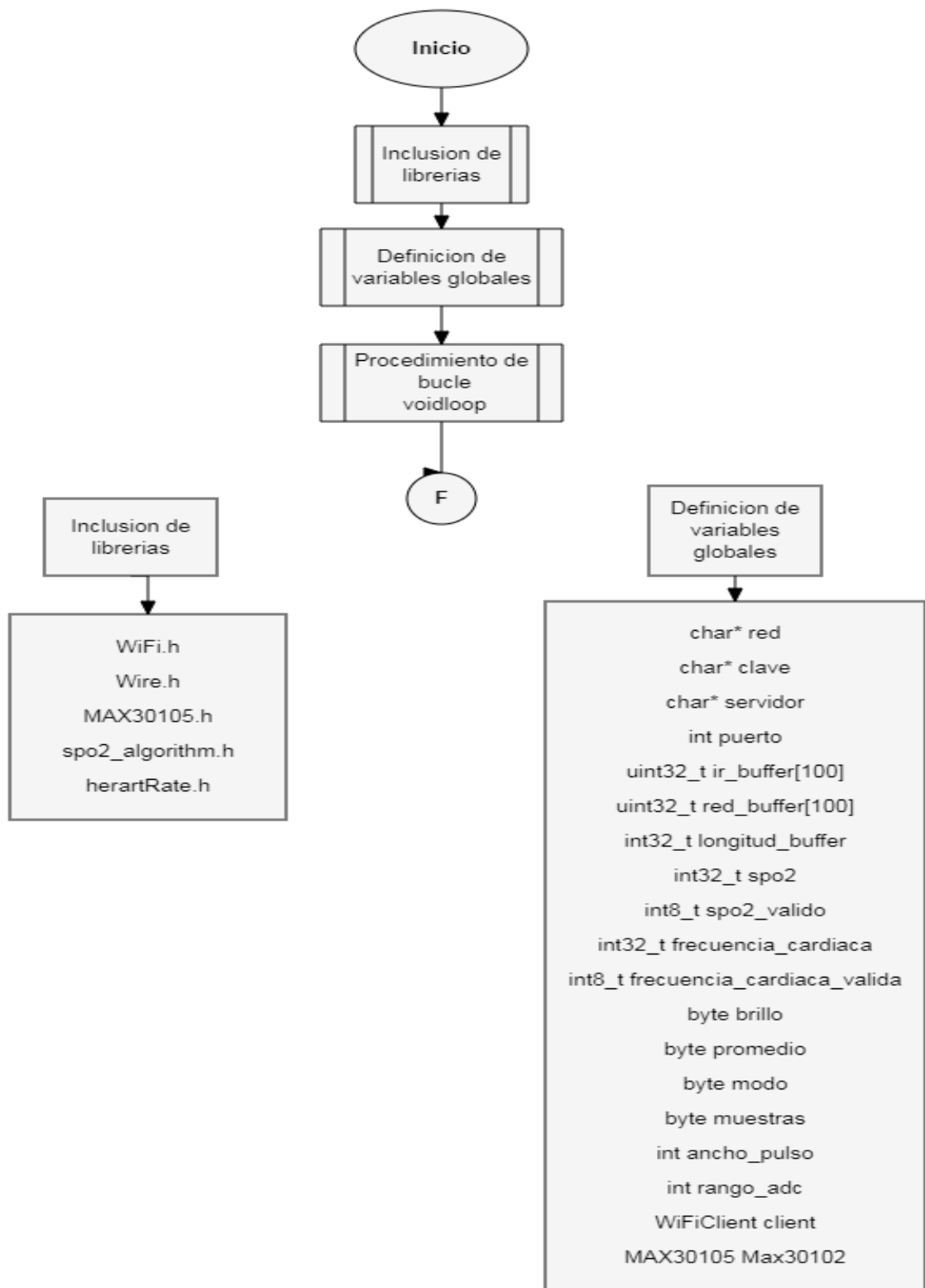


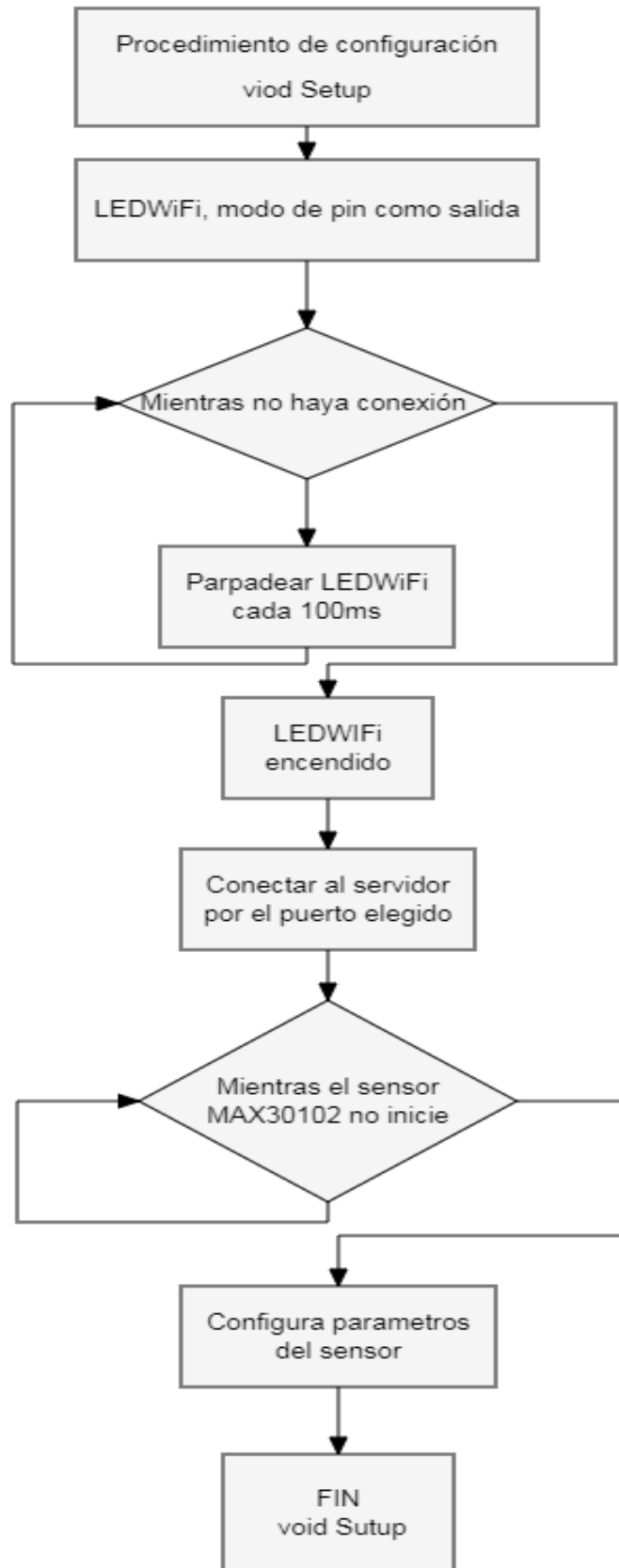
Figura 36: Recepción de datos en el PC, programa modo claro.

Fuente: Captura de pantalla

3.7 DIAGRAMA DE FLUJO

3.7.1 DIAGRAMA DEL PROGRAMA ARDUINO





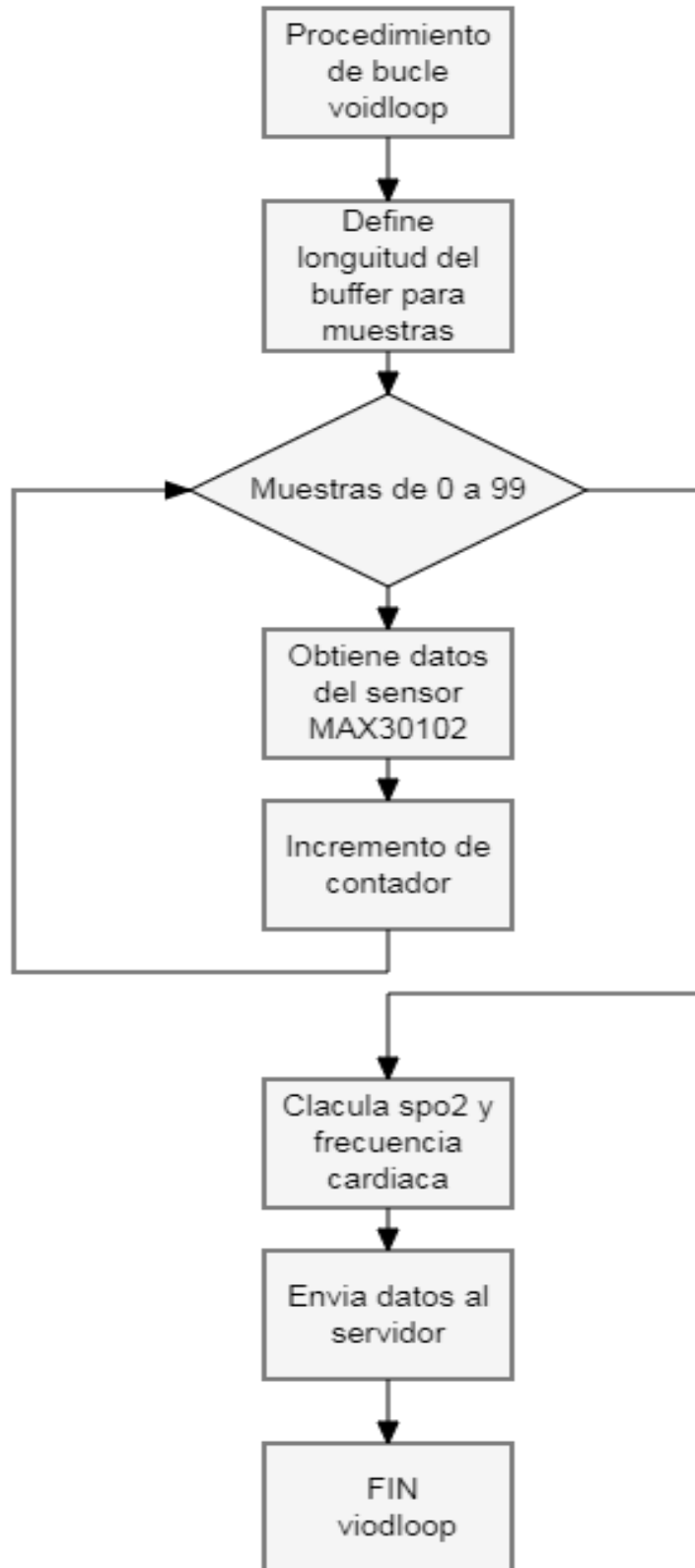
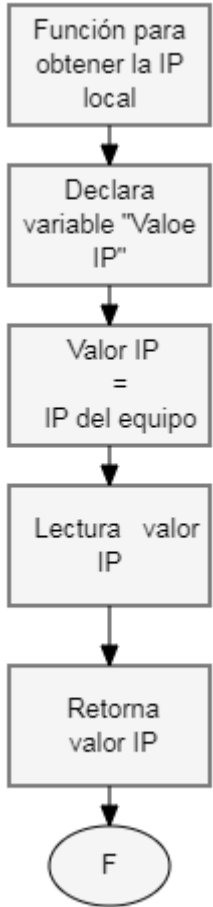
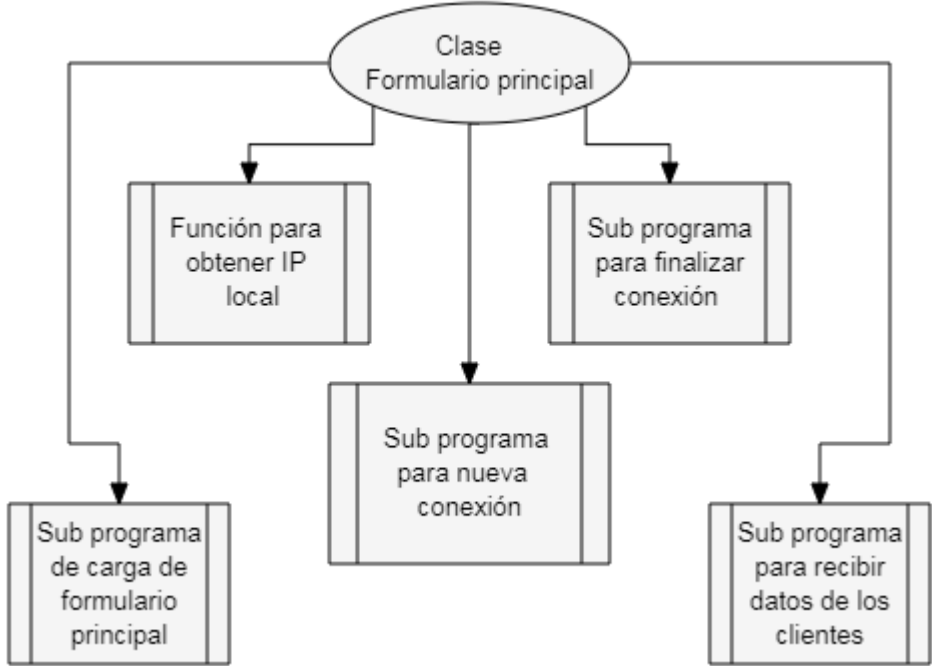


Figura 37: Diagrama de flujo del programa de Arduino.

Fuente: Diseño propio

3.7.2 DIAGRAMA DEL PROGRAMA DE LA PC



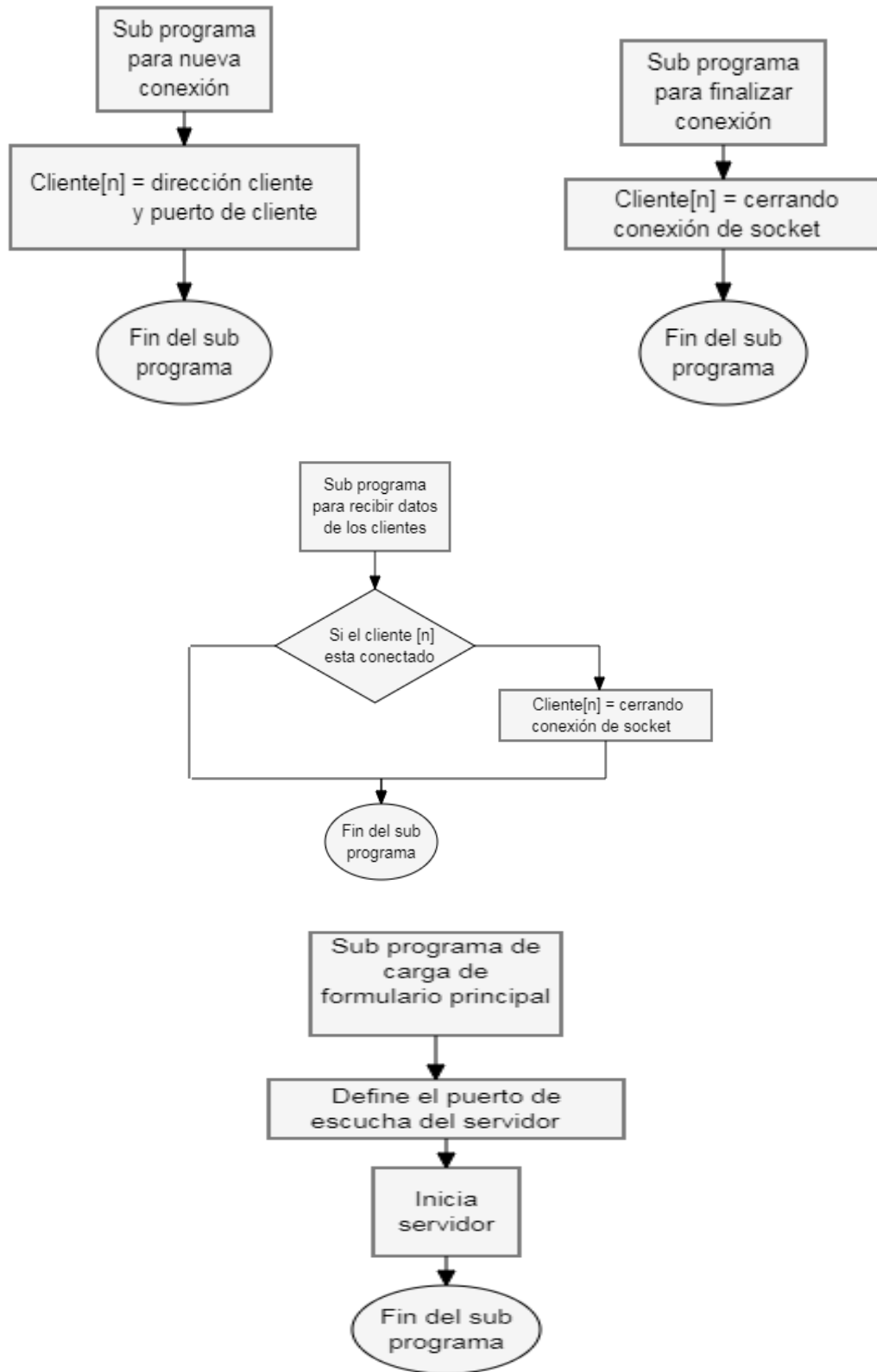


Figura 38: Diagrama de flujo del programa correspondiente a la PC.

Fuente: Diseño propio

CAPITULO IV

ANALISIS DE COSTOS

En este punto se detallará por medio de tablas los costos y detalles que implico el presente proyecto.

4.1 COSTO DE MATERIALES

CATEGORÍA	CANTIDAD	DESCRIPCIÓN	VALOR	PRECIO UNITARIO EN BS.
Resistor	2	¼ Vatio	4.7 k ohm	0,20
Capacitor	1	Electrolítico	100 uF	2
Diodo	1	LED	5mm	0,5
Módulos	1	Oxímetro	MAX30102	45
	1	Placa de desarrollo	ESP32	70
Conector	1	Conector SIL (espadín hembra)	-	4
Otros	1	1 m de cable de 4 hilos	-	2,50
	1	Placa de baquelita PCB	-	15
	1	Fuente de poder	5V	30
	1	Pasta de soldadura	-	5
	3	1 m de estaño	-	1,5
TOTAL				170,9

4.2 COSTOS DE OXÍMETROS COMERCIALES USADOS EN CENTROS DE SALUD

Algunos de los modelos comerciales, similares al prototipo a implementar están a la venta por precios similares; y teniendo en cuenta que en la tabla anteriormente elaborada se especifica el precio por unidad de los componentes, que es más elevado que si se realiza una compra de gran cantidad de cada elemento, el coste del diseño se reduciría. Por tanto, el diseño desarrollado cumple las expectativas dentro del mercado competitivo. Estos son algunos de los modelos que se pueden encontrar en las tiendas de distribución de equipos médicos.

MODELO	PRECIO (Bs)	DESCRIPCION GRAFICA
OXYM9000	350	
OXYM7500	920	
Nonin Go2	1100	
OXYM6100	1200	

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Realizado el proyecto se pueden dar las siguientes conclusiones:

- Se realizó el diseño un sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre para el centro de salud "Villa Armonía", de manera exitosa.
- Se pudo determinar las características técnicas del sistema de monitoreo masivo de pulso cardiaco y saturación de oxígeno en la sangre, esto se ve reflejado en el marco teórico del presente proyecto.
- Se estableció el tipo y las características del oxímetro en el proyecto, y se eligió el oxímetro MAX30102 el cual entrega los datos de manera digital.
- Se diseñó el firmware del microcontrolador, bajo el IDE de Arduino, y usando el microcontrolador ESP-32.
- Se efectuar el diseño el software para ajustes y configuración de parámetros, bajo el entorno de Visual Basic.NET usando los controles de Bunifu.
- La implementación del prototipo final y corrección de errores, se realizó haciendo pruebas previas las cuales llevaron a que el sistema se encuentre en optimo estado.

5.2 RECOMENDACIONES

En el periodo de pruebas que tuvo el proyecto se pudo calificar una serie de factores que podría interferir en el buen funcionamiento del prototipo, entre los cuales se presentaron, el ruido ambiente, ya que el dispositivo es muy sensitivo a interferencias electromagnéticas, también se presentó un inconveniente en los movimientos de los pacientes produciendo lecturas incorrectas. Es por ello que se recomienda evitar las siguientes situaciones que pueden evitar tener lecturas correctas que el dispositivo marque:

- ✓ **Intromisión o interferencia:** Esto a raíz de interferencias electromagnéticas que posee diferentes dispositivos electrónicos o equipos de alta potencia, por lo que

se recomienda tener el dispositivo en un área alejada de estos dispositivos o equipos que produzcan interferencias electromagnéticas.

- ✓ **Inquietud o movimiento:** La inquietud de paciente puede ocasionar errores de lectura del dispositivo, obteniendo datos incorrectos, por lo que se recomienda tranquilizar al paciente antes de realizar una medición en su pulso cardiaco.
- ✓ **Tiempo de Medida:** Se debe tomar en cuenta que el tiempo sugerido es de al menos un minuto para lograr obtener los datos suficientes y necesarios que necesita el sistema de monitoreo del paciente.
- ✓ **Uso de cosméticos:** El poder contar un área mínimamente limpia y libre de impurezas involucra verificar si el paciente cuenta con algún tipo de esmalte en área de la uña, debido a que colores específicos como ser; el color rojo, color negro y color verde pueden causar errores de lectura al momento de la medición.
- ✓ **Enfermedades de piel:** Es importante verificar la historia médica o notar si la piel del paciente tiene alguna pigmentación, herida o daño en la misma. Esto puede ocasionar que el dispositivo no pueda captar con facilidad el flujo sanguíneo.
- ✓ **Esmalte de uñas:** El color rojo, negro y verde, pueden causar errores en la lectura de la SpO₂, debido a que alteran la absorbancia lumínica e interfieren con las dos longitudes de onda.
- ✓ **Pigmentación de la piel:** La medición puede ser errónea cuando la pigmentación de la piel es oscura, debido a que el sensor no puede captar con facilidad el flujo sanguíneo.

BIBLIOGRAFIA

- Alex Rolando Poma Canaza. (2018). *Implementación de un dispositivo electrónico digital para la detección de frecuencia cardiaca y saturación de oxígeno con interfaz a ordenador y Smartphone*. Facultad de Tecnología de la Universidad Mayor de San Andrés.
- Santtiago Ventosinos Alonso. (2010). *Sistema optico autonomo para la medida de parametros biomedicos*. Universidad Plitecnica de Catalunya. España
- David López Serrano. (2017). *Diseño e Implementación de un Pulsioxímetro*. Universitat Politècnica de València.
- Luis Manuel Moreno Moruno. (2015). *Diseño e Implementación de un Dispositivo para la Detección de la Oxigenación en Sangre*. Departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla.
- Jorge Alarco Perez. (2015). *Medida de nivel de Saturacion de oxígeno en la sangre. Desarrollo de un pulsioximetro de bajo costo*. Escuela Tecnica Superior de Ingenieros de Telecomunicaciones en Madrid.
- Revista Ingenieria UC. (2016). *Diseño y construccion de un oximetro de Pulso*. Facultad de Ingenieria. Univerddidad de Carabobo
- Farid Manuel Medina Nuñez. (2019). *Construccion de un Oximetro de pulso Inalambrico*. Facultad de Ingenierias. Universidad Tecnologica de Bolivar
- Medline Plus. (2021). *Terapiacon oximetro*. Obtenido de <https://medlineplus.gov/spanish/oxygentherapy.html>
- Medline Plus. (2021). *Cardiopatía*. Obtenido de <https://medlineplus.gov/spanish/oxygentherapy.html>
- Medline Plus. (2021). *Sangre*. Obtenido de <https://medlineplus.gov/spanish/blood.html>
- Wikipedia, la enciclopedia libre. (2021). *La Sangre*. Obtenido de https://es.wikipedia.org/wiki/Sangre#Composici%C3%B3n_de_la_sangre
- Ministerio de Salud. (2021). *Composicion de La Sangre*. Obtenido de https://msal.gob.ar/disah/index.php?option=com_content&id=315&Itemid=39

- Wikipedia, la enciclopedia libre. (2020). *Oxímetro*. Obtenido de <https://es.wikipedia.org/wiki/Ox%C3%ADmetro>
- Centro de investigaciones, Óptica. (2020). *Oximetría*. Obtenido de <https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/606/1/15270.pdf>
- Educación Médica Continua. (2021). *Oxímetro de Pulso*. Obtenido de https://scielo.org.bo/scielo.php?script=sci_arttext&pid=S1024067520120002000
- Somib, (Sociedad Mexicana Biomédica). (2020). *Oximetría con Plemistógrafo*. Obtenido de <https://C:/Users/Usuario/Downloads/admin-gestora-de-la-revista-86-trabajos-libres.pdf>
- S. Cielo, (Sociedad Cielo). (2020). *Oxímetro de Pulso*. Obtenido de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S102406752012000200011
- Universidad Técnica de la Mixteca. (2020). *Medidor de Potencia Técnica*. Obtenido de <https://C:/Users/Usuario/Desktop/8130.pdf>
- Universidad Ciencia y Tecnología. (2020). *Señal de Pulso Arterial*. Obtenido de https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316482120050002000
- Wikipedia, la enciclopedia libre. (2020). *Lenguaje de alto nivel*. Obtenido de https://es.wikipedia.org/Lenguaje_de_alto_nivel/
- Wikipedia, la enciclopedia libre. (2021). *Visual Basic .NET*. Obtenido de https://es.wikipedia.org/wiki/Visual_Basic_.NET
- Desarrollo web. (2020). *Tipos de Datos en .NET*. Obtenido de <https://desarrolloweb.com/articulos/1388.php>
- Datagridview. (2021). *Que es y para qué sirve un Datagridview*. Obtenido de <https://envb.net/datagridview/>
- Wikipedia, la enciclopedia libre. (2022). *Microsoft Visual Studio*. Obtenido de https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- Microsoft Build. (2022). *Conozca la Familia de Visual*. Obtenido de <https://visualstudio.microsoft.com/es>
- Microsoft AppSource. (2022). *Bunifu Framework-Visual Studio*. Obtenido de <https://appsource.microsoft.com/eses/product/webapps/bunifutechnologiesltd.bunifudevtools?tab=overview>

Microsoft Documentos. (2022). *Familia de Soluciones de tipo Socket*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.socket?view=net-6.0#constructors>

Wikipedia, la enciclopedia libre. (2021). *Arduino*. Obtenido de <https://es.wikipedia.org/wiki/Arduino>

Aprendiendo con Arduino, Wordpress. (2021). *Ide de Arduino*. Obtenido de <https://aprendiendoarduino.wordpress.com/2017/06/18/ide-arduino-y-configuracion>

Wikipedia, la enciclopedia libre. (2020). *IPV4*. Obtenido de <https://es.wikipedia.org/wiki/IPV4>

CCM. (2021). *Protocolo TCP*. Obtenido de [https:// es.ccm.net/contents/281-protocolo-tcp#que-es-elprotocolo-tcp](https://es.ccm.net/contents/281-protocolo-tcp#que-es-elprotocolo-tcp) co

Wikipedia, la enciclopedia libre. (2020). *Wifi*. Obtenido de [https:// es.wikipedia.org/wiki/Wifi](https://es.wikipedia.org/wiki/Wifi)

Mas Adelante. (2020). *Definicion del Servidor*. Obtenido de [https:// www.masadelante.com/faqs/servidor](https://www.masadelante.com/faqs/servidor)

Wikipedia, la enciclopedia libre. (2020). *RS-232*. Obtenido de [https:// es.wikipedia.org/wiki/RS-232](https://es.wikipedia.org/wiki/RS-232)

Wikipedia, la enciclopedia libre. (2020). *Microprocesador*. Obtenido de [https:// es.wikipedia.org/wiki/Microprocesador](https://es.wikipedia.org/wiki/Microprocesador)

Wikipedia, la enciclopedia libre. (2020). *Microcontrolador*. Obtenido de <https://es.wikipedia.org/wiki/Microcontrolador>

Wikipedia, la enciclopedia libre. (2020). *ESP32* Obtenido de <https://es.wikipedia.org/wiki/ESP32>

Wikipedia, la enciclopedia libre. (2021). *Método científico*. Obtenido de https://es.wikipedia.org/wiki/M%C3%A9todo_cient%C3%ADfico

Easy Pulse Sensor, Version 1.1. (2021). *Sensor de pulso*. Obtenido de <http://www.embedded-lab.com/blog/easy-pulse-version-1-1-sensor-overview-part-1/>

Medical News Today, Foro de salud. (2022). *Nivel de oxigeno en la sangre*. Obtenido de <https://www.medicalnewstoday.com/search?q=oxigeno%20en%20la%20sangrel>

Arduino CC, Hadware. (2022). *Comunicación Serial Arduino*. Obtenido de <https://playground.arduino.cc/ArduinoNotebookTraduccion/Serial>

ADJUNTOS

CÓDIGO FUENTE DE LA APLICACIÓN DE CONFIGURACIÓN EN VB.NET

```
Imports System.Threading
Imports System.Net.Sockets
Imports System.Text
Imports System.Net

Public Class Servidor
    'Esta estructura permite guardar la información sobre un cliente
    Private Structure datosClienteConectado
        Public socketConexion As Socket          'Socket para mantener la conexión con cliente
        Public Thread As Thread                 'Hilo para mantener escucha con el cliente
        Public UltimosDatosRecibidos As String  'Últimos datos enviados por el cliente
    End Structure
    Private tcpLsn As TcpListener              'Para realizar la escucha de conexiones de
    clientes
    Private Clientes As New Hashtable()        'Datos de los clientes conectados
    Private tcpThd As Thread
    Private IDClienteActual As Net.IPEndPoint  'Último cliente conectado
    Private m_PuertoDeEscucha As String

    Public Event NuevaConexion(ByVal IDTerminal As Net.IPEndPoint)
    Public Event DatosRecibidos(ByVal IDTerminal As Net.IPEndPoint)
    Public Event ConexionTerminada(ByVal IDTerminal As Net.IPEndPoint)
    'Propiedad para establecer el puerto por el que se realizará la escucha
    Property Puerto() As Integer
        Get
            Puerto = m_PuertoDeEscucha
        End Get

        Set(ByVal Value As Integer)
            m_PuertoDeEscucha = Value
        End Set
    End Property
    'Procedimiento para establecer el servidor en modo escucha
    Public Sub IniciarEscucha()
        tcpLsn = New TcpListener(IPAddress.Any, Puerto)
        'Iniciar escucha
        tcpLsn.Start()
        'Crear hilo para dejar escuchando la conexión de clientes
        tcpThd = New Thread(AddressOf EsperarConexionCliente)
        tcpThd.Start()
    End Sub
    'Procedimiento para detener la escucha del servidor
    Public Sub DetenerEscucha()
        CerrarTodosClientes()
        tcpThd.Abort()
        tcpLsn.Stop()
    End Sub
    Public Function ObtenerDatos(ByVal IDCliente As Net.IPEndPoint) As String
        Dim InfoClienteSolicitado As datosClienteConectado
        'Obtengo la información del cliente solicitado
        InfoClienteSolicitado = Clientes(IDCliente)
        ObtenerDatos = InfoClienteSolicitado.UltimosDatosRecibidos
    End Function
    'Cierra la conexión de un cliente conectado
    Public Sub cerrarConexionCliente(ByVal IDCliente As Net.IPEndPoint)
        Dim InfoClienteActual As datosClienteConectado
        'Obtener información del cliente indicado
        InfoClienteActual = Clientes(IDCliente)
        'Cerrar conexión con cliente
        InfoClienteActual.socketConexion.Close()
    End Sub
End Class
```

```

End Sub
'Cerrar todas la conexión de todos los clientes conectados
Public Sub CerrarTodosClientes()
    Dim InfoClienteActual As datosClienteConectado
    'Cerrar conexión de todos los clientes
    For Each InfoClienteActual In Clientes.Values
        Call cerrarConexionCliente(InfoClienteActual.socketConexion.RemoteEndPoint)
    Next
End Sub
'Enviar mensaje a cliente indicado
Public Sub enviarMensajeCliente(ByVal IDCliente As Net.IPEndPoint, ByVal Datos As String)
    Dim Cliente As datosClienteConectado
    'Obtener información del cliente al que se enviará el mensaje
    Cliente = Clientes(IDCliente)
    'Enviar mensaje a cliente
    Cliente.socketConexion.Send(Encoding.ASCII.GetBytes(Datos))
End Sub
'Enviar mensaje a todos los clientes conectados al servidor
Public Sub enviarMensajeTodosClientes(ByVal Datos As String)
    Dim Cliente As datosClienteConectado
    For Each Cliente In Clientes.Values
        enviarMensajeCliente(Cliente.socketConexion.RemoteEndPoint, Datos)
    Next
End Sub
'Procedimiento que inicia la espera de la conexión de un cliente
'para ello inicia un hilo (thread)
Private Sub EsperarConexionCliente()
    Dim datosClienteActual As datosClienteConectado
    While True
        'Se guarda la información del cliente cuando se recibe la conexión
        'Quedará esperando la conexión de un nuevo cliente
        datosClienteActual.socketConexion = tcpLsn.AcceptSocket()
        'Con el IDClienteActual se identificará al cliente conectado
        IDClienteActual = datosClienteActual.socketConexion.RemoteEndPoint
        'Crear un hilo para que quede escuchando los mensajes del cliente
        datosClienteActual.Thread = New Thread(AddressOf LeerSocket)
        'Agregar la información del cliente conectado al array
        SyncLock Me
            Clientes.Add(IDClienteActual, datosClienteActual)
        End SyncLock
        'Generar evento NuevaConexion
        RaiseEvent NuevaConexion(IDClienteActual)
        'Iniciar el hilo que escuchará los mensajes del cliente
        datosClienteActual.Thread.Start()
    End While
End Sub
'Procedimiento para leer datos enviados por el cliente
Private Sub LeerSocket()
    Dim IDReal As Net.IPEndPoint 'ID del cliente que se va a escuchar
    Dim Recibir() As Byte 'Array donde se guardarán los datos que lleguen
    Dim InfoClienteActual As datosClienteConectado 'Datos del cliente conectado
    Dim Ret As Integer = 0
    IDReal = IDClienteActual
    InfoClienteActual = Clientes(IDReal)
    While True
        If InfoClienteActual.socketConexion.Connected Then
            Recibir = New Byte(100) {}
            Try
                'Esperar a que lleguen un mensaje desde el cliente
                Ret = InfoClienteActual.socketConexion.Receive(Recibir,
                    Recibir.Length, SocketFlags.None)
                If Ret > 0 Then
                    'Guardar mensaje recibido
                    InfoClienteActual.UltimosDatosRecibidos =
Encoding.ASCII.GetString(Recibir)
                    Clientes(IDReal) = InfoClienteActual
                    'Generar el evento DatosRecibidos
                    'para los datos recibidos
                    RaiseEvent DatosRecibidos(IDReal)
                End If
            Catch
            End Try
        End If
    End While
End Sub

```

```

        Else
            'Generar el evento ConexionTerminada
            'de finalización de la conexión
            RaiseEvent ConexionTerminada(IDReal)
            Exit While
        End If
    Catch e As Exception
        If Not InfoClienteActual.socketConexion.Connected Then
            'Generar el evento ConexionTerminada
            'de finalización de la conexión
            RaiseEvent ConexionTerminada(IDReal)
            Exit While
        End If
    End Try
End If
End While
Call CerrarThread(IDReal)
End Sub
'Procedimiento para cerrar el hilo (thread)
Private Sub CerrarThread(ByVal IDCliente As Net.IPEndPoint)
    Dim InfoClienteActual As datosClienteConectado
    'Finalizar el hilo (thread) iniciado
    ' encargado de escuchar al cliente
    InfoClienteActual = Clientes(IDCliente)
    Try
        InfoClienteActual.Thread.Abort()
    Catch e As Exception
        SyncLock Me
            'Eliminar el cliente del array
            Clientes.Remove(IDCliente)
        End SyncLock
    End Try
End Sub
End Class

```

```

////////////////////////////////////
////                                UNIVERSIDAD MAYOR DE SAN ANDRÉS                                ////
////                                FACULTAD DE TECNOLOGÍA                                    ////
////                                CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES                ////
////////////////////////////////////
////                                PROYECTO DE GRADO                                        ////
////                                ////                                                    ////
////                                SISTEMA DE MONITOREO MASIVO DE PULSO CARDIACO            ////
////                                Y SATURACIÓN DE OXÍGENO EN LA SANGRE PARA EL            ////
////                                CENTRO DE SALUD "VILLA ARMONÍA"                          ////
////////////////////////////////////
////                                POR : ISAAC ALBARO APAZA RAMIREZ                            ////
////                                TUTOR : LIC. EDDER TOMÁS JURADO MOYA                    ////
////////////////////////////////////
////                                Archivo : Oximetro.ino                                    ////
////////////////////////////////////
////                                Inclusión de librerías                                    ////
////////////////////////////////////
#include <WiFi.h> //Incluye la librería para la comunicación WiFi ////
#include <Wire.h> //Incluye la librería para el protocolo I2C ////
#include <MAX30105.h> //Incluye la librería para el sensor MAX30102 ////
#include <spo2_algorithm.h> //Incluye la librería que obtiene valor de SP02 ////
#include <heartRate.h> //Incluye la librería que obtiene valor de BPM ////
////////////////////////////////////
////                                Definiciones                                            ////
////////////////////////////////////
#define LED_WIFI 2 //Define al pin 2 como LED_WIFI ////
////////////////////////////////////
////                                Definición de variables globales                        ////
////////////////////////////////////
const char* red = "B890-950D"; //Nombre de red WiFi ////
const char* clave = "D053E3A9"; //Contraseña de red WiFi ////
const char* servidor = "192.168.1.5"; //Dirección IP del servidor ////
const int puerto = 9999; //Puerto de escucha del servidor ////
////
uint32_t ir_buffer[100]; //Define el buffer de muestras de IR ////
uint32_t red_buffer[100]; //Define el buffer de muestras de Red ////
int32_t longitud_buffer; //Longitud del buffer = 100 ////
////
int32_t spo2; //Obtiene el valor de spo2 ////
int8_t spo2_valido; //Bandera que detecta dato valido ////
int32_t frecuencia_cardiaca; //Obtiene el valor de bpm ////
int8_t frecuencia_cardiaca_valida; //Bandera que detecta dato valido ////
////
byte brillo = 60; //Opciones: 0=apagado a 255=50mA ////
byte promedio = 4; //Opciones: 1, 2, 4, 8, 16, 32 ////
byte modo = 2; //Opciones: 1 = Rojo, 2 = Rojo+IR, 3 = Rojo+IR+Verde ////
byte muestras = 100; //Opciones: 50, 100, 200, 400, 800, 1000, 1600, 3200 ////
int ancho_pulso = 411; //Opciones: 69, 118, 215, 411 ////
int rango_adc = 4096; //Opciones: 2048, 4096, 8192, 16384 ////
////
WiFiClient client; //Instancia el Objeto client ////
MAX30105 Max30102; //Instancia el objeto Max30102 ////
////////////////////////////////////
////                                Procedimiento de configuración                        ////
////////////////////////////////////
void setup()
{
pinMode(LED_WIFI, OUTPUT); //Configura LED_WIFI como salida ////
WiFi.begin(red, clave) //Inicia la conexión a la red WiFi ////
while(WiFi.status() != WL_CONNECTED) //Mientras no se logre conectar ////
{
digitalWrite(LED_WIFI, HIGH); //Pone en Alto a LED_WIFI ////
delay(100); //Espera 100 ms ////
digitalWrite(LED_WIFI, LOW); //Pone en Bajo a LED_WIFI ////
delay(100); //Espera 100 ms ////
}
digitalWrite(LED_WIFI, HIGH); //Pone en Alto a LED_WIFI ////
client.connect(servidor, puerto); //Inicia la conexión con el servidor ////
while(!Max30102.begin(Wire, I2C_SPEED_FAST))//Mientras no inicie el sensor////
{
}
}

```



```

} ////////////////////////////////////////////////////
//Configura parametros del sensor ////////////////////////////////////////////////////
Max30102.setup(brillo, promedio, modo, muestras, ancho_pulso, rango_adc); ////////////////////////////////////////////////////
} ////////////////////////////////////////////////////
// Procedimiento de configuración ////////////////////////////////////////////////////
// Procedimiento de configuración ////////////////////////////////////////////////////
void loop() ////////////////////////////////////////////////////
{ ////////////////////////////////////////////////////
  longitud_buffer = 100; //Cantidad de muestras ////////////////////////////////////////////////////
  for(int i = 0; i < longitud_buffer; i++) //for de 0 a 99 para muestras ////////////////////////////////////////////////////
  { ////////////////////////////////////////////////////
    while (Max30102.available() == false) //Mientras no haya dato ////////////////////////////////////////////////////
    { ////////////////////////////////////////////////////
      Max30102.check(); //Verifica dato en el sensor ////////////////////////////////////////////////////
    } ////////////////////////////////////////////////////
    red_buffer[i] = Max30102.getRed(); //Obtiene El dato del led rojo ////////////////////////////////////////////////////
    ir_buffer[i] = Max30102.getIR(); //Obtiene el dato del led Ir ////////////////////////////////////////////////////
    Max30102.nextSample(); //Toma la siguiente muestra ////////////////////////////////////////////////////
  } ////////////////////////////////////////////////////
  //Calcula el Spo2 y frecuencia cardiaca ////////////////////////////////////////////////////
  maxim_heart_rate_and_oxygen_saturation(ir_buffer, longitud_buffer, ////////////////////////////////////////////////////
  red_buffer, &spo2, &spo2_valido, &frecuencia_cardiaca, ////////////////////////////////////////////////////
  &frecuencia_cardiaca_valida); ////////////////////////////////////////////////////
} ////////////////////////////////////////////////////
// Procedimiento de configuración ////////////////////////////////////////////////////

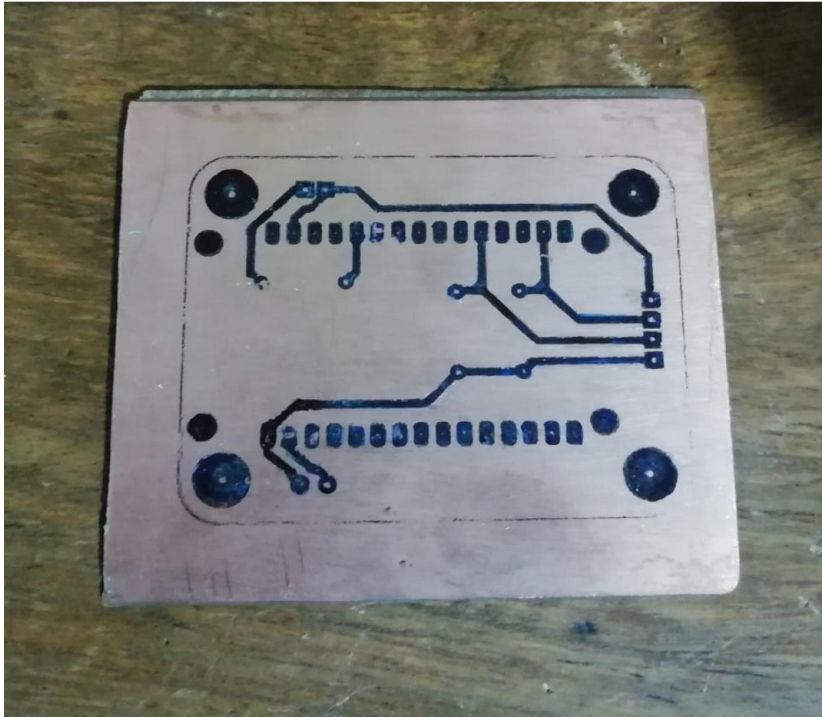
```

ELABORACIÓN DEL PCB CORRESPONDIENTE

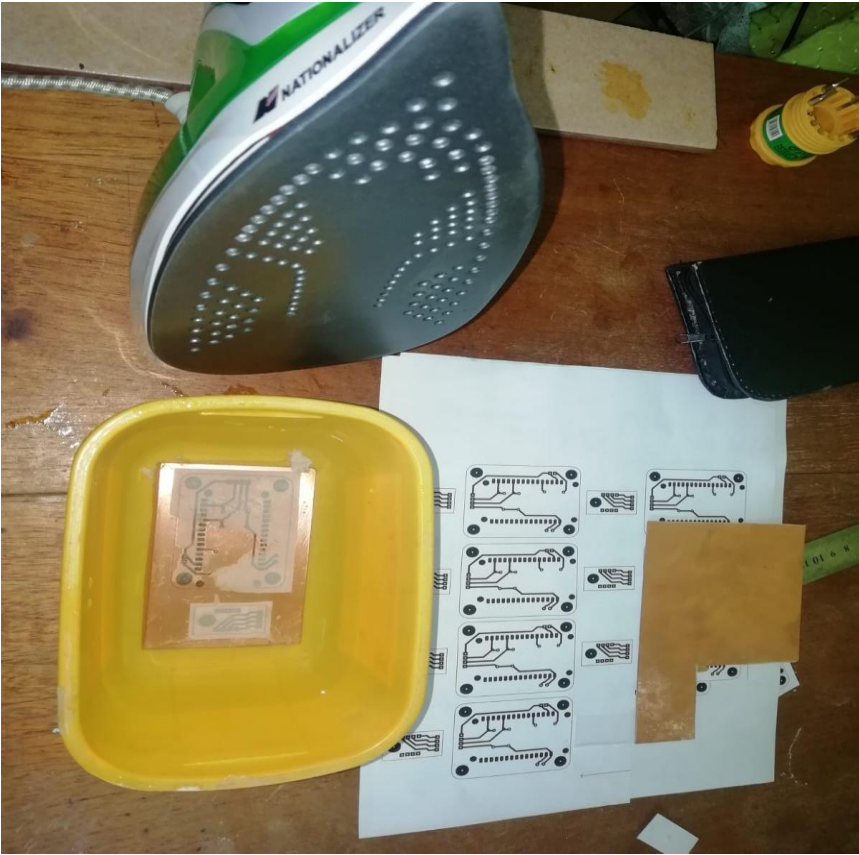
Método del planchado, uso de la trasferencia térmica.



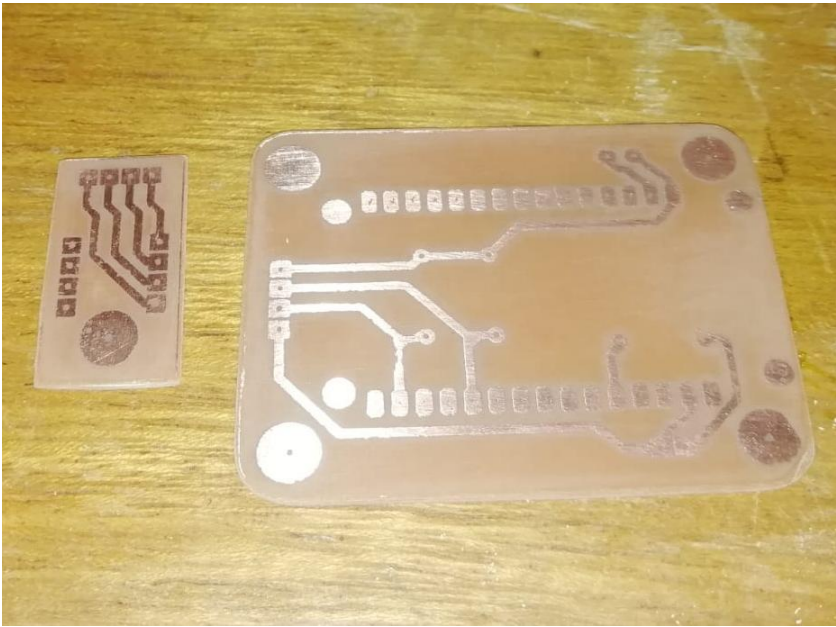
Resultado de la trasferencia térmica.

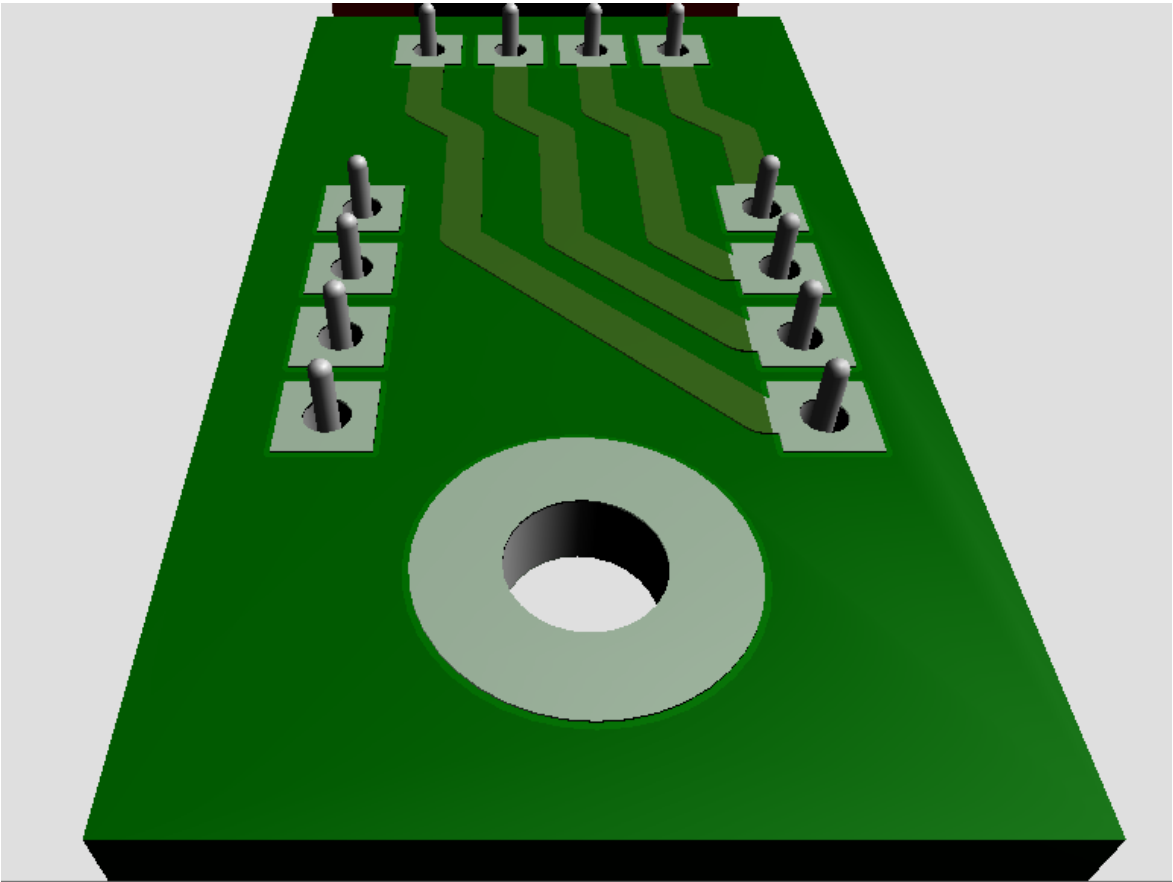
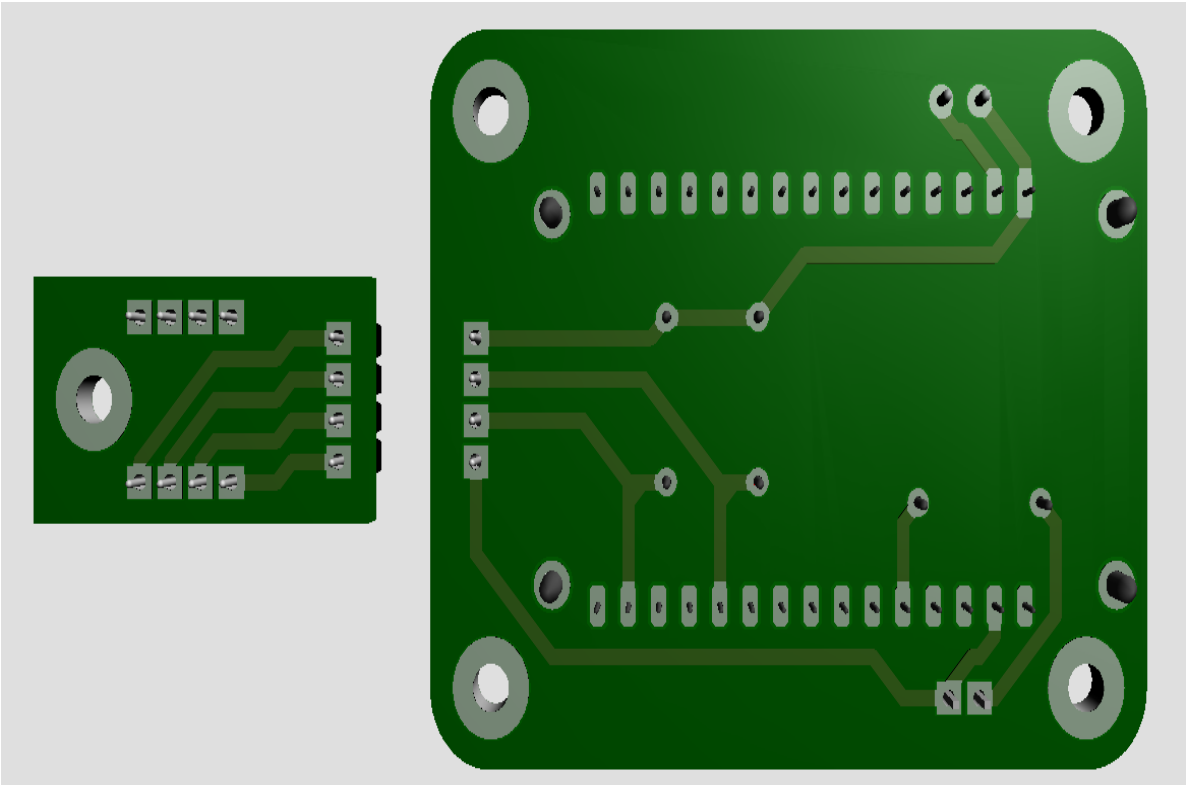


Concluido con el planchado, el circuito reposa en un recipiente de agua.

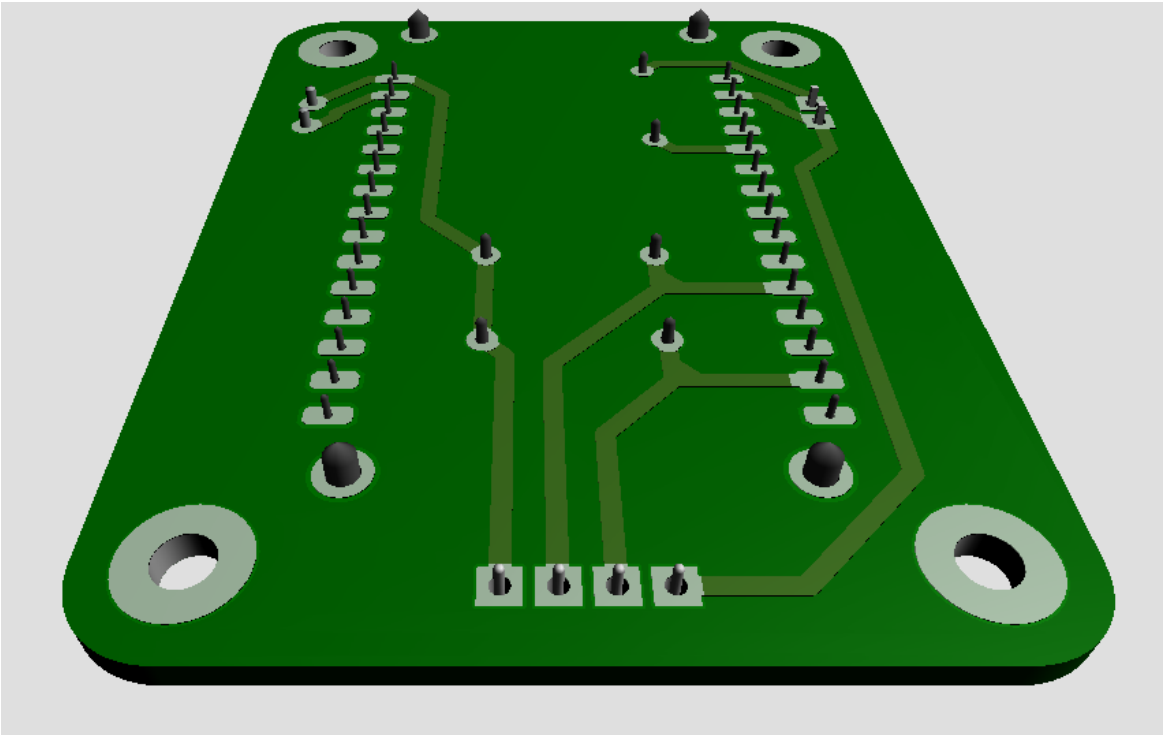


Resultado del proceso del quemado de la placa PCB en acido férrico.

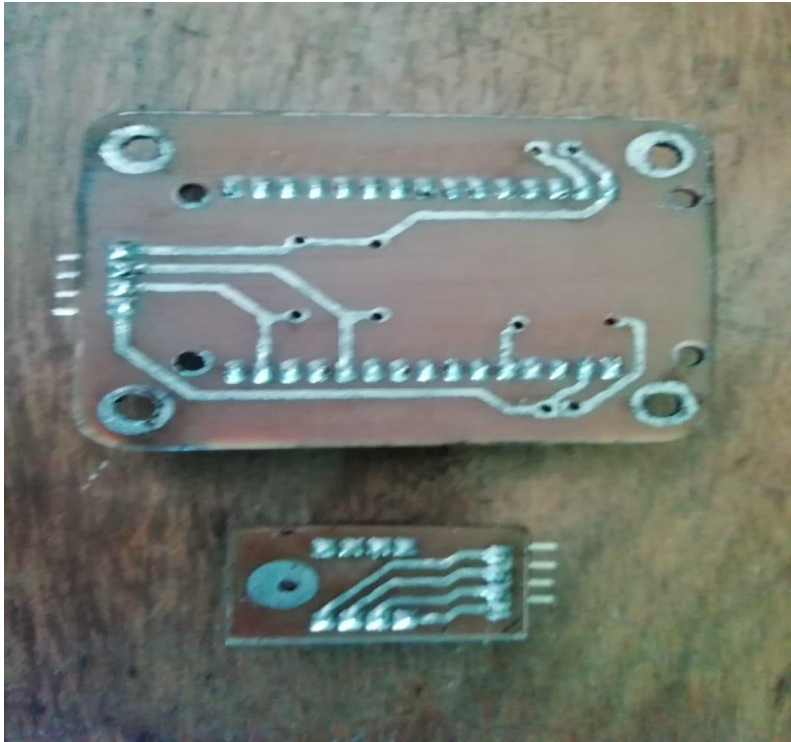




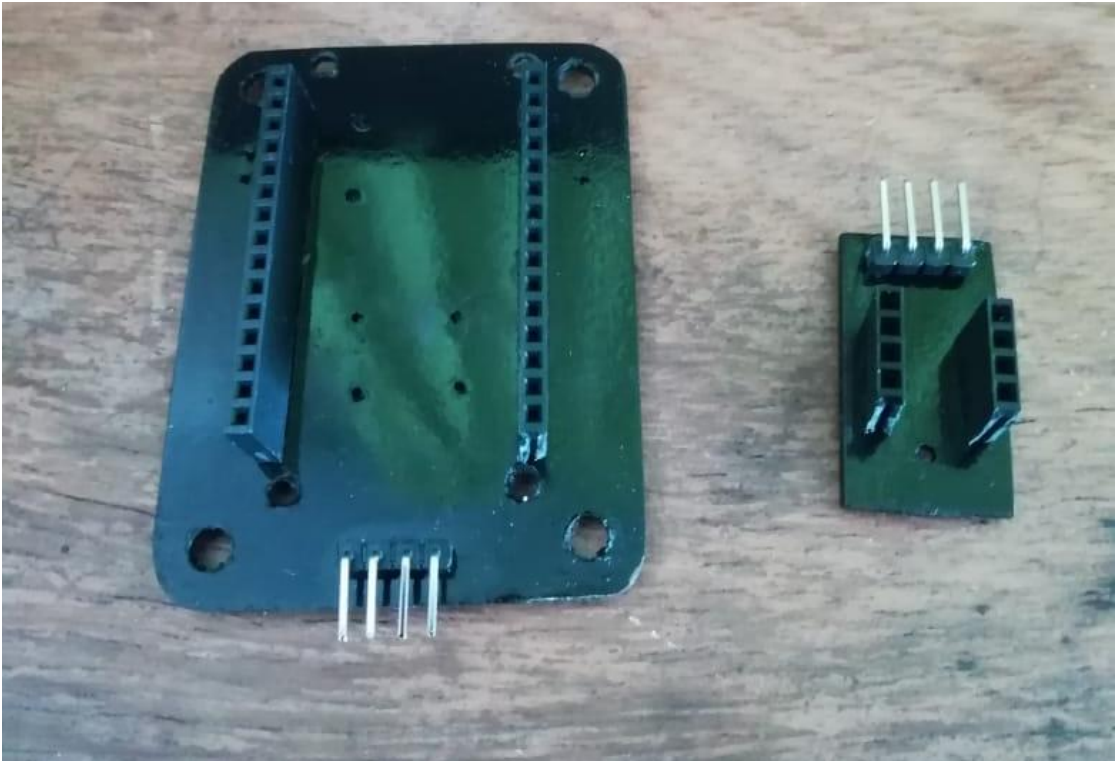
Simulación 3D del diseño del dispositivo.



Perforado y soldado de componentes a la placa PCB.



Resultado final del proceso del diseño y quemado de la placa PCB.



Prototipo final.

